Enzo Lectures

Mike Norman, Matt Turk
Laboratory for Computational Astrophysics
UC San Diego

	Morning	Afternoon
Mon.	Introduction to Enzo	
Tue.	1. Setting Up and Running Enzo 2. Enzo Projects	Introduction to YT
Wed.	Basic Enzo Algorithms	Lab session
Thu.	Applications to First Stars, First Galaxies, and Reionization	Lab session
Fri.	What’s New in Enzo 2.0?	Q & A

Basic Enzo Algorithms

Greg Bryan (Columbia University)
gbryan@astro.columbia.edu

Topics

I. Hydrodynamics
   - PPM
   - ZEUS
II. AMR
   - Timestepping
   - Projection
   - Flux correction
III. Gravity
   - Root grid
   - Subgrids
IV. Particles
V. Chemistry & Cooling
   - Multispecies

I. Hydrodynamics

Show PPM movies

Fluid Equations - grid::SolveHydroEquations

Mass conservation
$$\frac{\partial \rho}{\partial t} + \frac{1}{a}\mathbf{v}\cdot\nabla\rho = -\frac{1}{a}\rho\nabla\cdot\mathbf{v}$$

Momentum conservation
$$\frac{\partial \mathbf{v}}{\partial t} + \frac{1}{a}(\mathbf{v}\cdot\nabla)\mathbf{v} = -\frac{\dot{a}}{a}\mathbf{v} - \frac{1}{a\rho}\nabla p - \frac{1}{a}\nabla\phi,$$

Energy conservation
$$\frac{\partial E}{\partial t} + \frac{1}{a}\mathbf{v}\cdot\nabla E = -\frac{\dot{a}}{a}(3\frac{p}{\rho} + \mathbf{v}^2) - \frac{1}{a\rho}\nabla\cdot(p\mathbf{v}) - \frac{1}{a}\mathbf{v}\cdot\nabla\phi + \Gamma - \Lambda.$$

$$E = e + \frac{1}{2}\mathbf{v}^2,$$

Ideal Gas EOS
$$e = p/\left[(\gamma - 1)\,\rho\right],$$

Self-gravity
$$\nabla^2\phi = \frac{4\pi G}{a}(\rho_{total} - \rho_0).$$

Field names: Density, Pressure, TotalEnergy, InternalEnergy, Velocity1, Velocity2, Velocity3

grid class: accessing the fields – grid.h

- In grid class:
  - BaryonFields[] – array of pointers to each field
    - Fortran (row-major) ordering within each field
  - GridRank – dimensionality of problem
  - GridDimensions[] – dimensions of this grid
  - GridStartIndex[] – Index of first "active" cell (usually 3)
    - First (and last) three cells are ghost or boundary zones

```
int DensNum = FindField(Density, FieldType, NumberOfBaryonFields);
int Vel1Num = FindField(Velocity1, FieldType, NumberOfBaryonFields);

for (k = GridStartIndex[2]; k <= GridEndIndex[2]; k++) {
 for (j = GridStartIndex[1]; j <= GridEndIndex[1]; j++) {
  for (i = GridStartIndex[0]; i <= GridEndIndex[0]; i++) {
   BaryonField[Vel1Num][GINDEX(i,j,k)] *= BaryonField[DensNum][GINDEX(I,j,k)];
  }
 }
}
```

Enzo file name convention

- General C++ routines:
  - Routine name: EvolveLevel(…)
  - In file: EvolveLevel.C
  - One routine per file!
- grid methods:
  - Routine name: grid::MyName(…)
  - In file: Grid_MyName.C
- Fortran routines:
  - Routine name: intvar(…)
  - In file: intvar.src
    - .src is used because routine is fed first through C preprocessor

PPM Solver: grid::SolvePPM_DE

- HydroMethod = 0
- PPM: e.g. mass conservation equation
  - Flux conservative form:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v}{\partial x} = 0$$

$$\rho_j^n = \rho(x_j, t^n)$$

  - In discrete form:

Mass flux across j+1/2 boundary

$$\rho_j^{n+1} = \rho_j^n + \Delta t \left(\frac{\overline{\rho}_{j+1/2}\overline{v}_{j+1/2} - \overline{\rho}_{j-1/2}\overline{v}_{j-1/2}}{\Delta x_j} \right)$$

  - How to compute mass flux?
- Note: multi-dimensions handled by operating splitting
  - grid::xEulerSweep.C, grid::yEulerSweep.C, grid::zEulerSweep.C

Grid::SolvePPM_DE

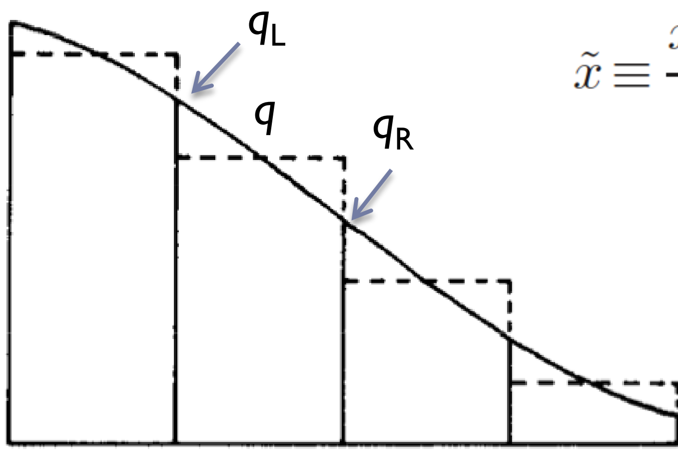
```
// Update in x-direction
for (k = 0; k < GridDimension[2]; k++) {
    if (this->xEulerSweep(k, NumberOfSubgrids, SubgridFluxes,
                          GridGlobalStart, CellWidthTemp, GravityOn,
                          NumberOfColours, colnum) == FAIL) {
        fprintf(stderr, "Error in xEulerSweep.  k = %d\n", k);
        ENZO_FAIL("");
    }
} // ENDFOR k

// Update in y-direction
for (i = 0; i < GridDimension[0]; i++) {
    if (this->yEulerSweep(i, NumberOfSubgrids, SubgridFluxes,
                          GridGlobalStart, CellWidthTemp, GravityOn,
                          NumberOfColours, colnum) == FAIL) {
      fprintf(stderr, "Error in yEulerSweep.  i = %d\n", i);
      ENZO_FAIL("");
    }
  } // ENDFOR i
```

PPM: 1D hydro update: grid::xEulerSweep

- Copy 2D slice out of cube
- Compute pressure on slice (pgas2d)
- Calculate diffusion/steepening coefficients (calcdiss)
- Compute Left and Right states on each cell edge (inteuler)
- Solve Reimann problem at each cell edge (twoshock)
- Compute fluxes of conserved quantities at each cell edge (euler)
- Save fluxes for future use
- Return slice to cube

PPM: reconstruction: inteuler

- Piecewise parabolic representation:



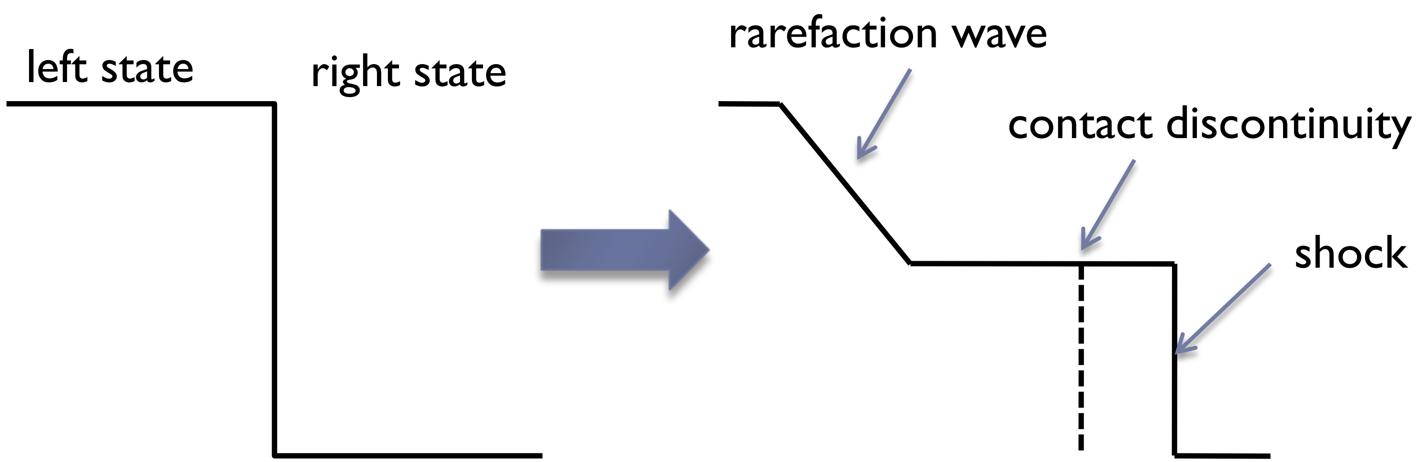
$$q_j(x) = q_{L,j} + \tilde{x}(\Delta q_j + q_{6,j}(1 - \tilde{x})),$$

$$\tilde{x} \equiv \frac{x - x_{j-1/2}}{\Delta x_j}, \qquad x_{j-1/2} \leq x \leq x_{j+1/2}.$$

- Coefficients (Δq and q_6) computed with mean q and q_L, q_R.
- For smooth flow (like shown above), this is fine, but can cause a problem for discontinuities (e.g. shocks)
- q_L, q_R are modified to ensure monotonicity (no *new* extrema)

PPM: Godunov method: twoshock

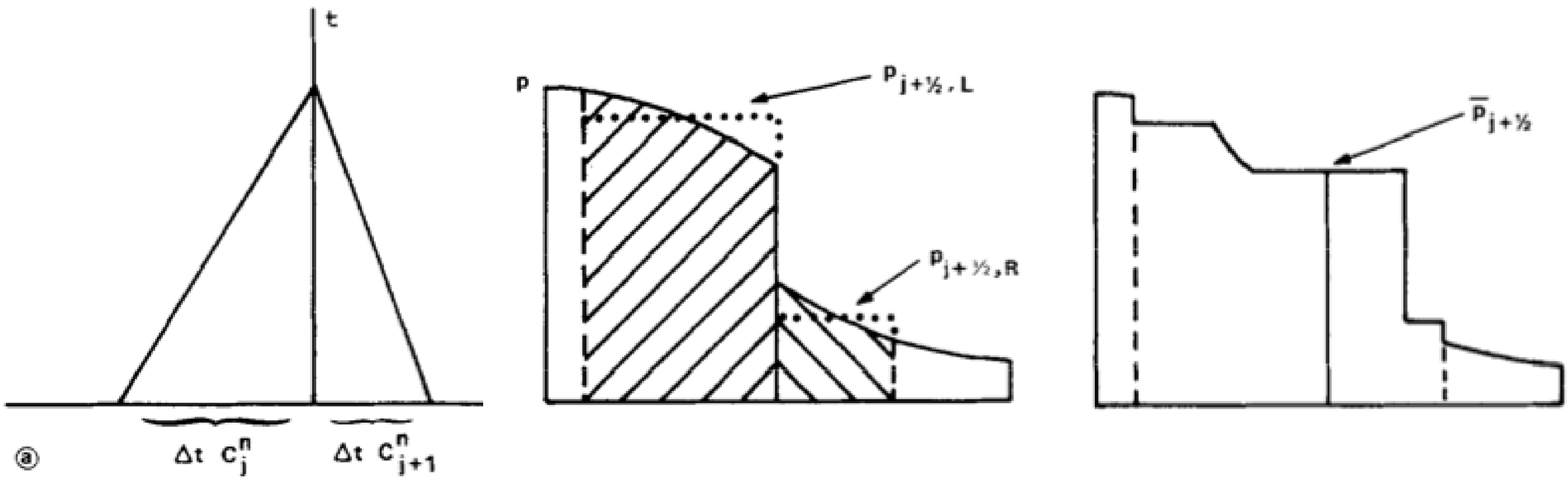
- To compute flux at cell boundary, take two initial constant states and then solve Riemann problem at interface



- Given solution, can compute flux across boundary
- Advantage: correctly satisfies jump conditions for shock

PPM: Godunov method: inteuler, twoshock

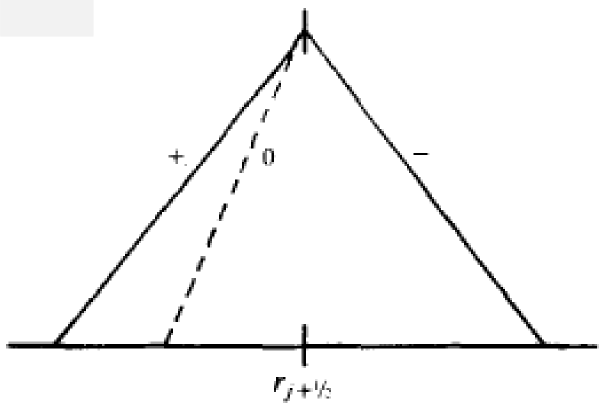
- For PPM, compute left and right states by averaging over characteristic region (causal region for time step Δt)



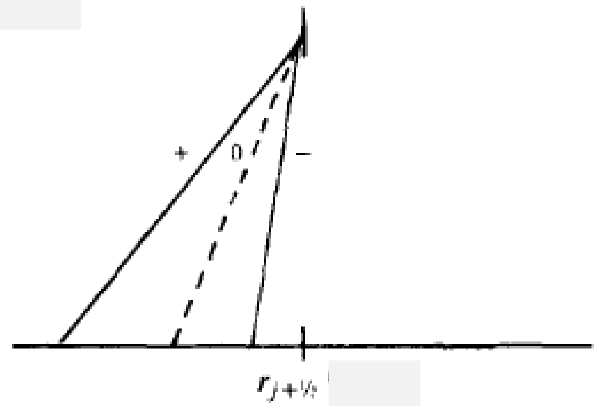
- Average left and right regions become constant regions to be feed into Riemann solver (twoshock).

PPM: Eulerian corrections: euler

- Eulerian case more complicated because cell edge is fixed.
  - Characteristic region for fixed cell more complicated:



SUBSONIC CASE



SUPERSONIC CASE

  - Note that solution is not known ahead of time so two-step procedure is used (see Collela & Woodward 1984 for details)

Difficulty with very high Mach flows

- PPM is flux conservative so natural variables are mass, momentum, *total* energy
- Internal energy (e) computed from total energy (*E*):

$$e = E - \frac{1}{2}\mathbf{v^2}$$

- Problem can arise in *very* high Mach flows when *E* >> e
  - e is difference between two large numbers
- Not important for flow dynamics since *p* is negligible
  - But can cause problems if we want accurate temperatures since *T* α e

Dual Energy Formalism:
grid::ComputePresureDualEnergyFormalism

- Solution: Also evolve equation for internal energy:

$$\frac{\partial e}{\partial t} + \frac{1}{a}\mathbf{v} \cdot \nabla e = \frac{p}{a\rho}\nabla \cdot \mathbf{v}$$

- Select energy to use depending on ratio e/E:

$$p = \begin{cases} \rho(\gamma - 1)(E - \mathbf{v}^2/2), & (E - \mathbf{v}^2/2)/E > \eta_1; \\ \rho(\gamma - 1)e, & (E - \mathbf{v}^2/2)/E < \eta_1. \end{cases}$$

- Select with DualEnergyFormalism = 1
- Use when v/c_s > ~20
- Q: Why not just use e?
  - A: Equation for e is not in conservative form (source term).
  - Source term in internal energy equation causes diffusion

Zeus Solver: grid::ZeusSolver

- Traditional finite difference method
  - Artificial viscosity (see Stone & Norman 1992)
- HydroMethod = 2
- Source step: ZeusSource
  - Pressure (and gravity) update: $$v_j^{n+a} = v_j^n - \frac{\Delta t}{\Delta x_j} \frac{p_j^n - p_{j-1}^n}{(\rho_j^n + \rho_{j-1}^n)/2}$$
  - Artificial viscosity: $$v_j^{n+b} = v_j^{n+a} - \frac{\Delta t}{\Delta x_j} \frac{q_j^{n+a} - q_{j-1}^{n+a}}{(\rho_j^n + \rho_{j-1}^n)/2}$$ $$q_j = \begin{cases} Q_{\rm AV}\rho_j(v_{j+1} - v_j)^2 & \text{if}(\mathrm{v_{j+1} - v_j}) < 0 \\ 0 & \text{otherwise} \end{cases}$$
  - Compression heating: $$e_j^{n+c} = e_j^{n+b} \left(\frac{1 - (\Delta t/2)(\gamma - 1)(\nabla \cdot \mathbf{v})_j}{1 + (\Delta t/2)(\gamma - 1)(\nabla \cdot \mathbf{v})_j} \right)$$

Zeus Solver: grid::ZeusSolver

- **Transport step:** Zeus_xTransport

e.g. $$\rho_j^{n+d} = \rho_j^n - \frac{\Delta t}{\Delta x}(v_{j+1/2}^{n+c}\rho_{j+1/2}^{*} - v_{j-1/2}^{n+c}\rho_{j-1/2}^{*})$$

  - Note conservative form (transport part preserves mass)
  - Note v_{j+1} is face-centered so is really at cell-edge, but density needs to be interpolated. Zeus uses an upwinded van Leer (linear) interpolation:

$$q_j(x) = q_{L,j} + \tilde{x}(\Delta q_j$$

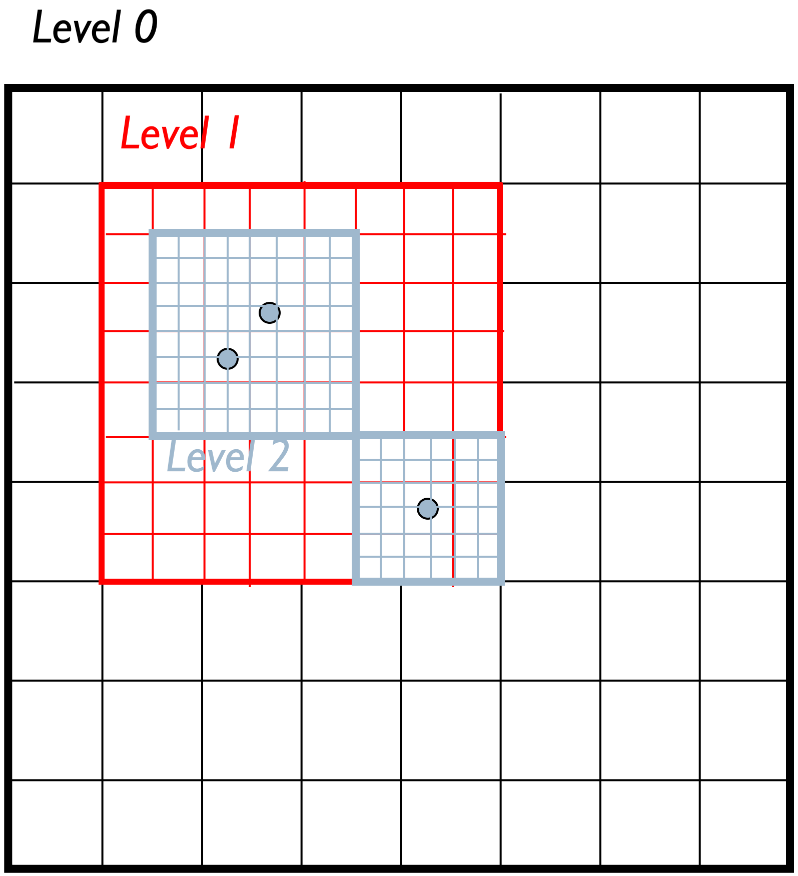
  - Similarly for momentum and energy (and y and z)
    - Zeus_yTransport, Zeus_zTransport

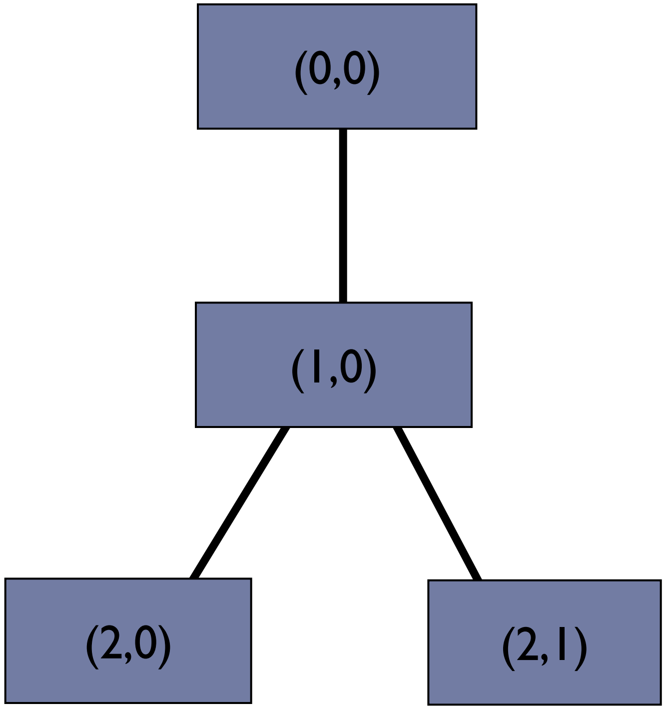
Zeus Solver: grid::ZeusSolver

- PPM is more accurate, slower but Zeus is faster and more robust.
  - PPM often fails ("dnu < 0" error) when fast cooling generates large density gradients.
  - Try out new hydro solvers in Enzo 2.0!
- Implementation differences with PPM:
  - Internal energy equation only
    - In code, TotalEnergy field is really internal energy (ugh!)
  - Velocities are face-centered
    - BaryonField[Vel1Num][GINDEX(i,j,k)] really "lives" at i-1/2

II. Block Structured AMR

Grid Hierarchy Data Structure





AMR: EvolveHierarchy

- Root grid NxNxN, so Δx = DomainWidth/N
- Level L defined so Δx = DomainWidth/($N2^L$)
- Starting with level 0, grid advanced by Δt
  - Main loop of EvolveHierarchy looks (roughly) like this:

```
InitializeHierarchy
While (Time < StopTime)
begin
  dt = ComputeTimeStep(0)
  EvolveLevel(0, dt)
  Time = Time + dt
  CheckForOutput(Time)
end
```

  - EvolveLevel does the heavy lifting

Time Step: grid::ComputeTimeStep

- Timestep on level L is minimum of constraints over all level L grids:

$$\Delta t_{hydro} = min\left(\kappa_{hydro}\frac{a\Delta x}{c_s + |v_x|}\right)_L$$

κ_{hydro} CourantSafetyFactor

$$\Delta t_{dm} = min\left(\kappa_{dm}\frac{a\Delta x}{v_{dm,x}}\right)_L ,$$

κ_{dm} ParticleCourantSafetyFactor

$$\Delta t_{exp} = f_{exp}\left(\frac{a}{\dot{a}}\right) ,$$

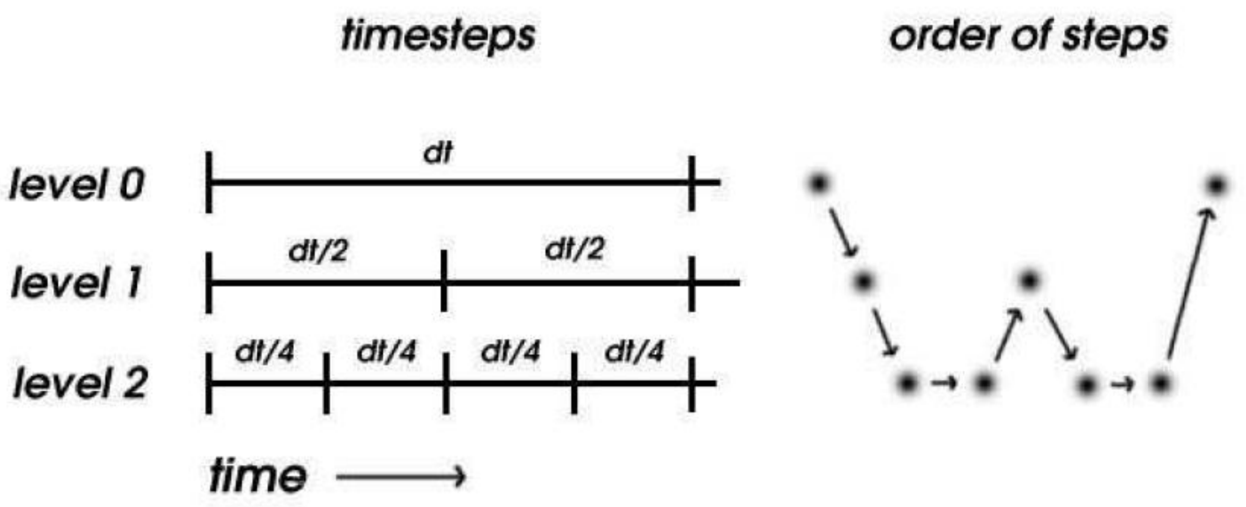
f_{exp} MaximumExpansionFactor

$$\Delta t_{accel} = min\left(\sqrt{\frac{\Delta x}{\vec{g}}}\right)_L$$

- + others (e.g. MHD, FLD, etc.)

AMR: EvolveLevel

- Levels advanced as follows:



  - Timesteps may not be integer ratios
    - (Diagram assumes Courant condition dominates and sound speed is constant so: dt α Δx)
  - This algorithm is defined in EvolveLevel

Advance grids on level: EvolveLevel

- The logic of EvolveLevel is given (roughly) as:

```
EvolveLevel(level)
begin
  SetBoundaryValues
  while (Time < ParentTime)
  begin
    dt = ComputeTimeStep(level)
    SolveHydroEquations(dt)
    Time = Time + dt
    SetBoundaryValues
    EvolveLevel(level+1, dt)
    FluxCorrection
    Projection
    RebuildHierarchy(level+1)
  end
end
```

Already talked about this. (dt = ComputeTimeStep(level), SolveHydroEquations(dt), Time = Time + dt)

recursive → EvolveLevel(level+1, dt)

Next, we'll talk about these (SetBoundaryValues, EvolveLevel(level+1, dt), FluxCorrection, Projection, RebuildHierarchy(level+1))

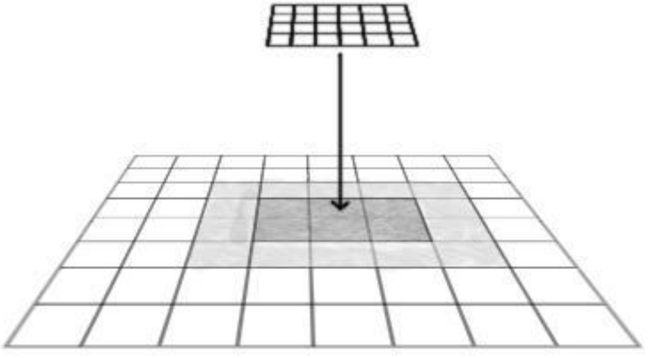
BC's: SetBoundaryConditions

- Setting "ghost" zones around outside of domain
  - grid::SetExternalBoundaryValues
  - Choices: reflecting, outflow, inflow, periodic
  - Only applied to level 0 grids (except periodic)
- Otherwise, two step procedure:
  - Interpolate ghost (boundary) zones from level L-1 grid
    - grid::InterpolateBoundaryFromParent
      - Linear interpolation in time (OldBaryonFields)
      - Spatial interpolation controlled by InterpolationMethod
        - SecondOrderA recommended, default (3D, linear in space, monotonic)
  - Copy ghost zones from sibling grids
    - grid::CheckForOverlap and grid::CopyZonesFromGrid

Projection: grid::ProjectSolutionToParentGrid

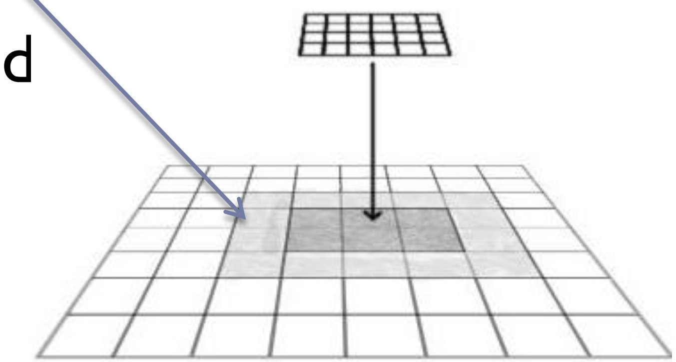
- Structured AMR produces redundancy:
  - coarse and fine grids cover same region
- Need to restore consistency
- Correct coarse cells once grids have all reach the same time:

$$q^{\text{coarse}} = r^{-d} \sum q_{i,j,k}^{\text{fine}}$$



Flux Correction: grid::CorrectForRefinedFluxes

- Mismatch of fluxes occurs around boundary of fine grids
  - Coarse cell just outside boundary used coarse fluxes but coarse cell inside used fine fluxes
- Both fine and coarse fluxes saved from hydro solver



$$q^{\text{coarse}} = \tilde{q}^{\text{coarse}} - \Delta t \left(F^{\text{coarse}} - \sum_{j,k} F_{j,k}^{fine} \right)$$

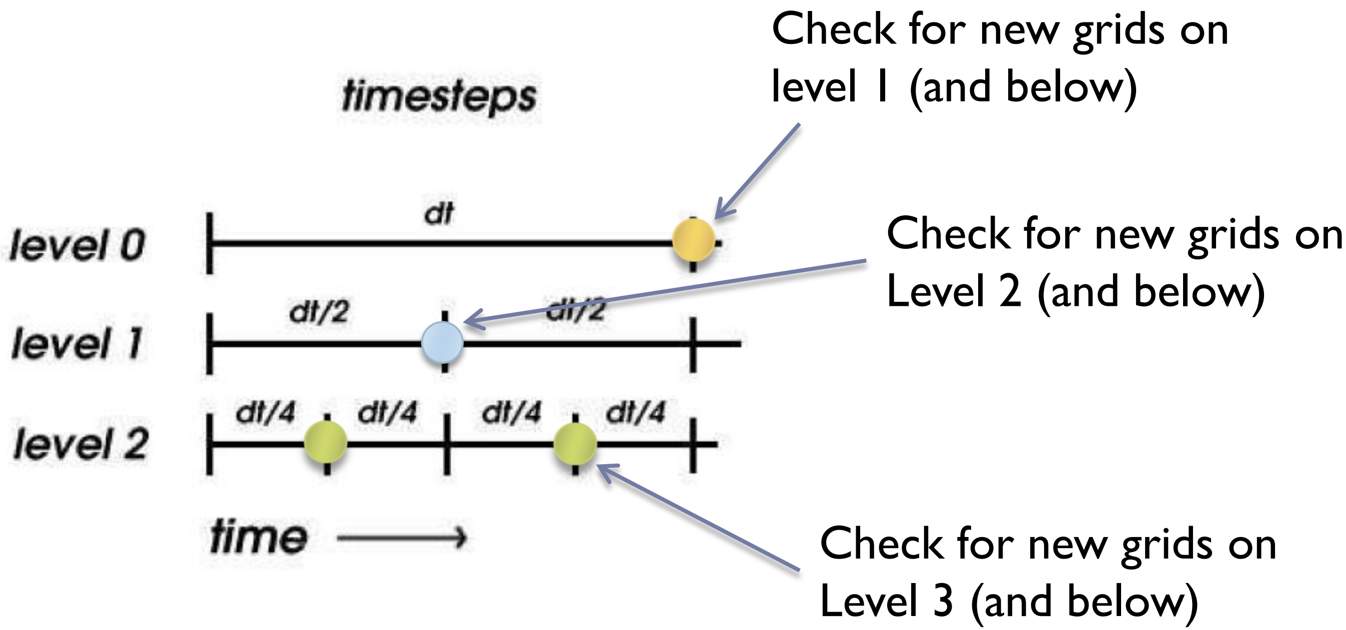
Uncorrected coarse value

Coarse flux across boundary

Sum of fine fluxes Over 4 (in 3D) abutting fine cells

Rebuilding the Hierarchy: RebuildHierarchy

- Need to check for cells needing more refinement



Refinement Criteria – grid::SetFlaggingField

- Many ways to flag cells for refinement
  - CellFlaggingMethod =

```
1 - refine by slope
2 - refine by baryon mass
3 - refine by shocks
4 - refine by particle mass
6 - refine by Jeans length
7 - refine if cooling time < cell width/sound speed
11 - refine by resistive length
12 - refine by defined region "MustRefineRegion"
13 - refine by metallicity
```

- Then rectangular grids must be chosen to cover all flagged cells with minimum "waste"
  - Done with machine vision technique
    - Looks for edges (inflection points in number of flagged cells)
  - ProtoSubgrid class

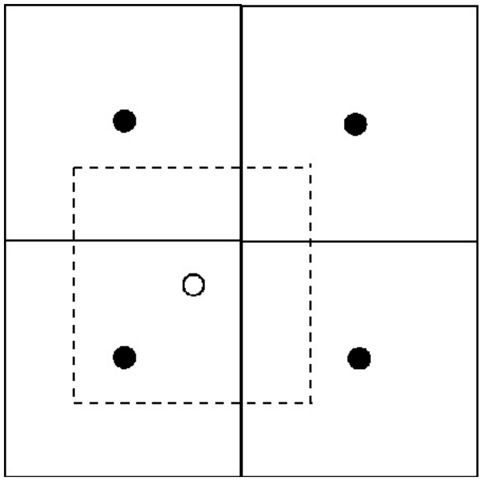
III. Gravity

Self-Gravity (SelfGravity = 1)

- Solve Poisson equation
- PrepareDensityField
  - BaryonField[Density] copied to GravitatingMassField
  - Particle mass is deposited in 8 nearest cells (CIC)
    - Particle position advanced by ½ step
    - DepositParticleMassField
- Root grid (level 0):
  - Potential solved with FFT

$$\tilde{\phi}(k) = G(k)\tilde{\rho}(k).$$

    - ComputePotentialFieldLevelZero
  - Potential differenced to get acceleration
    - grid::ComputeAccelerationField



Self-Gravity

- Subgrids:
  - Potential interpolated to boundary from parent
    - Grid::PreparePotentialField
  - Each subgrid then solves Poisson equation using multigrid
    - Grid::SolveForPotential
  - Note: this has two issues:
    - Interpolation errors on boundary can propagate to fine levels
      - Generally only an issue for steep potentials (point mass)
      - Ameliorated by having 6 ghost zones for gravity grid
    - Subgrids can have inconsistent potential gradients across boundary
      - Improved by copying new boundary conditions from siblings and resolving the Poisson equation (PotentialIterations = 4 by default)
    - More accurate methods in development

Other Gravitational sources – grid::ComputeAccelerationFieldExternal

- Can also add fixed potential:
  - UniformGravity – constant field
  - PointSourceGravity – single point source
  - ExternalGravity – NFW profile

IV. Particles

N-body dynamics

- Particles contribute mass to GravitatingMassField
- Particles accelerated by AccelerationField
  - Interpolated from grid (from 8 nearest cells)
- Particles advanced using leapfrog

$$x^{n+1/2} = x^n + (\Delta t/2) v^n$$
$$v^{n+1} = v^n + \Delta t a^{n+1/2}$$
$$x^{n+1} = x^{n+1/2} + (\Delta t/2) v^{n+1}$$

  - grid::ComputeAccelerations
- Particles stored in the locally most-refined grid
  - ParticlePosition, ParticleVelocity, ParticleMass
- Tracer particles (massless) also available

IV. Chemistry and Cooling

Chemistry

- Follows multiple species and solve rate equations

$$\frac{\partial \rho_i}{\partial t} + \frac{1}{a}\mathbf{v} \cdot \nabla \rho_i = -\frac{1}{a}\rho_i \nabla \cdot \mathbf{v} + \sum_j \sum_l k_{jl}(T)\rho_j \rho_l + \sum_j I_j \rho_j$$

  - MultiSpecies = 1: H, H+, He, He+, He++, e-
  - MultiSpecies = 2: adds H_2, H_2+, H-
  - MultiSpecies = 3: adds D, D+ and HD
  - grid:SolveRateEquations
    - (or grid::SolveRateAndCoolEquations if RadiativeCooling > 0)
- Rate equations solved using backwards differencing formula (BDF) with sub-cycles to prevent > 10% changes
  - Works well as long as chemical timescale not really short

Radiative Cooling – grid::SolveRadiativeCooling

- RadiativeCooling = 1
- Two modes:
  - MultiSpecies = 0
    - Equilibrium cooling table (reads file cool_rates.in)
    - Sub-cycles so that De < 10% in one cooling step
  - MultiSpecies > 1
    - Computes cooling rate self-consistently from tracked-species
    - MetalCooling = 1: adds metal cooling from Glover & Jappsen (2007)
    - MetalCooling = 2: adds metal cooling from Raymond-Smith code
    - MetalCooling = 3: Cloudy Cooling table (Smith, Sigurdsson & Abel 2008)
- RadiationFieldType > 0
  - Add predefined radiative heating and ionization

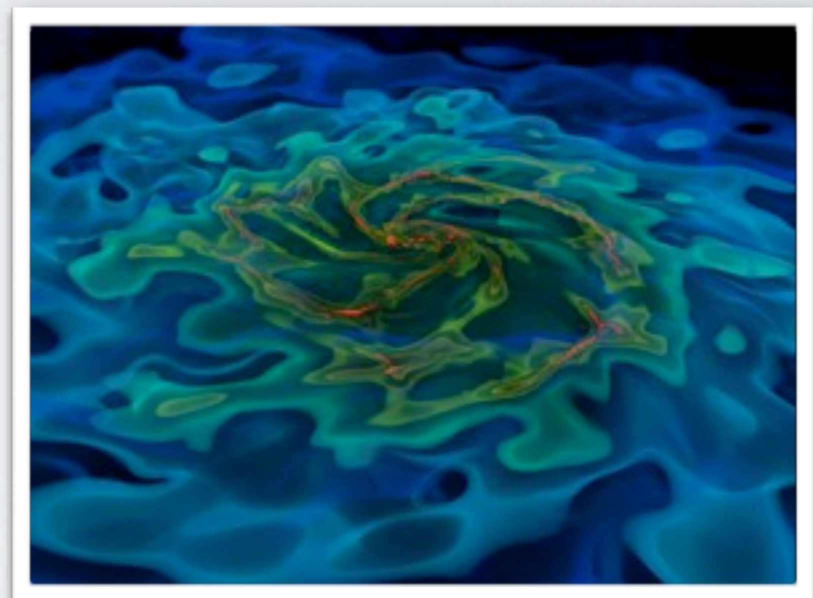
Star Formation

- Work in progress – many modes
- StarParticleCreation > 0
  - turns on and selects method (1-9)
  - For more details, see web page
- StarParticleFeedback > 0
  - Only valid for methods 1, 2, 7 and 8

More Physics in 2.0

- See talks tomorrow!

Galaxy Formation on ENZO with Properly Modeled Stars and MBHs

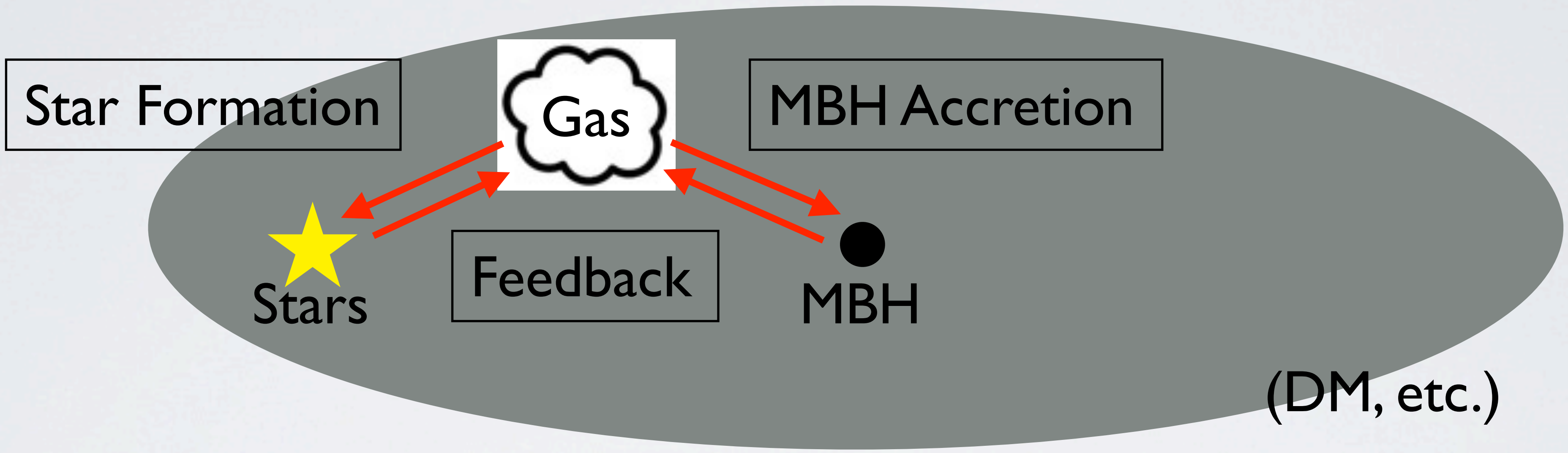


Ji-hoon Kim (KIPAC/Stanford)

Collaborators: John Wise(Princeton), Marcelo Alvarez(CITA),
Matthew Turk(UCSD), Tom Abel(Stanford)

Outline

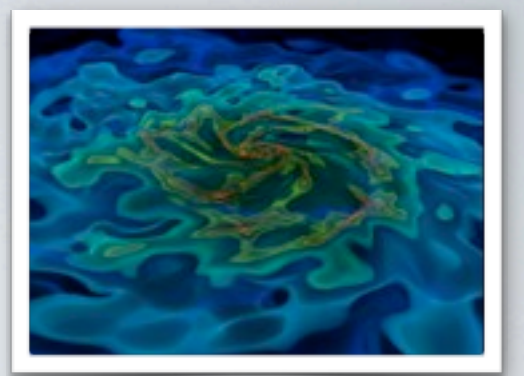
- Key Components to Understand and Simulate Galaxies



- Modeling the Physics of Galaxy Formation with Stars and MBHs As Best As You Can in ENZO

- Simulation Set-ups and Early Results

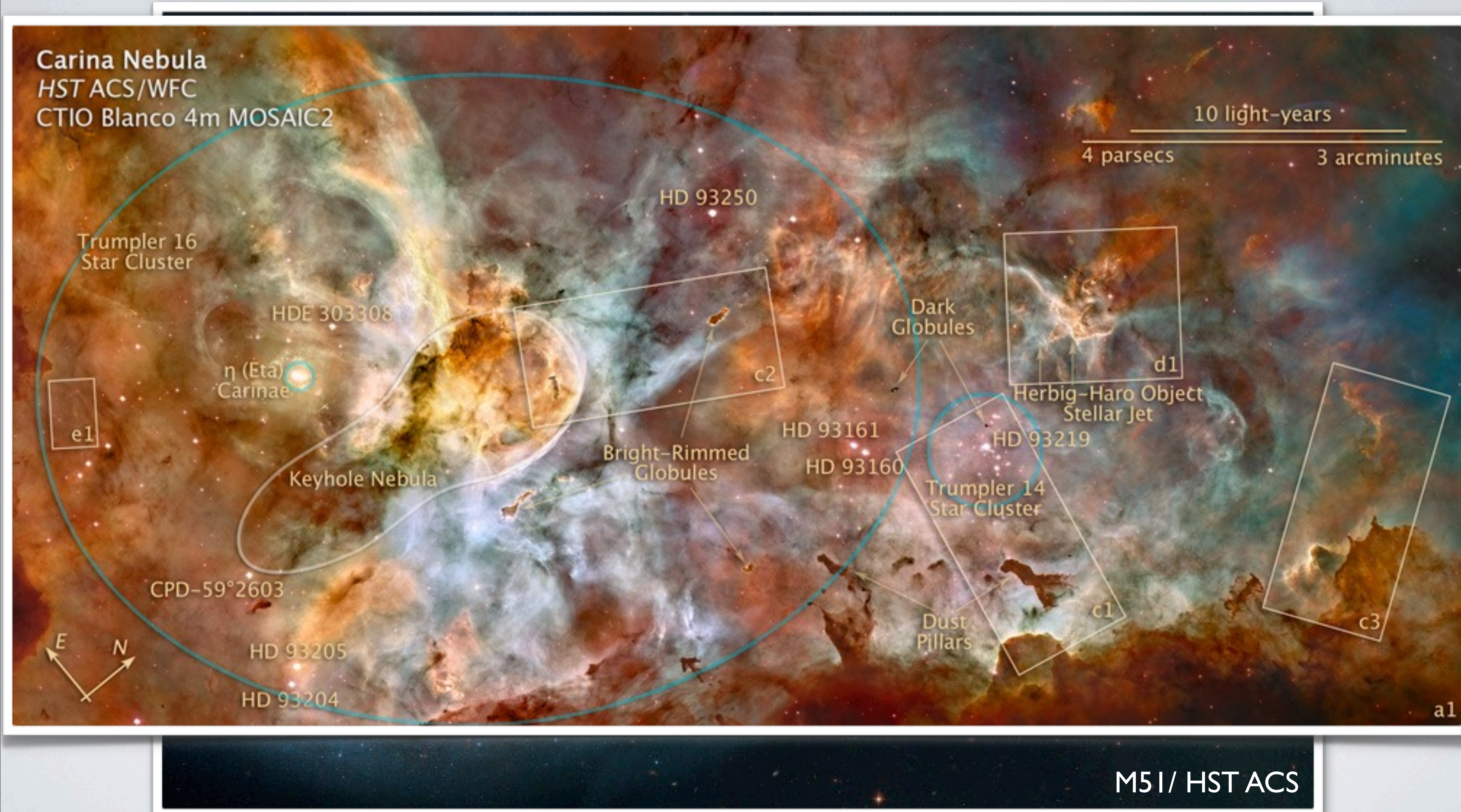
PART I



[Star Formation and Feedback]

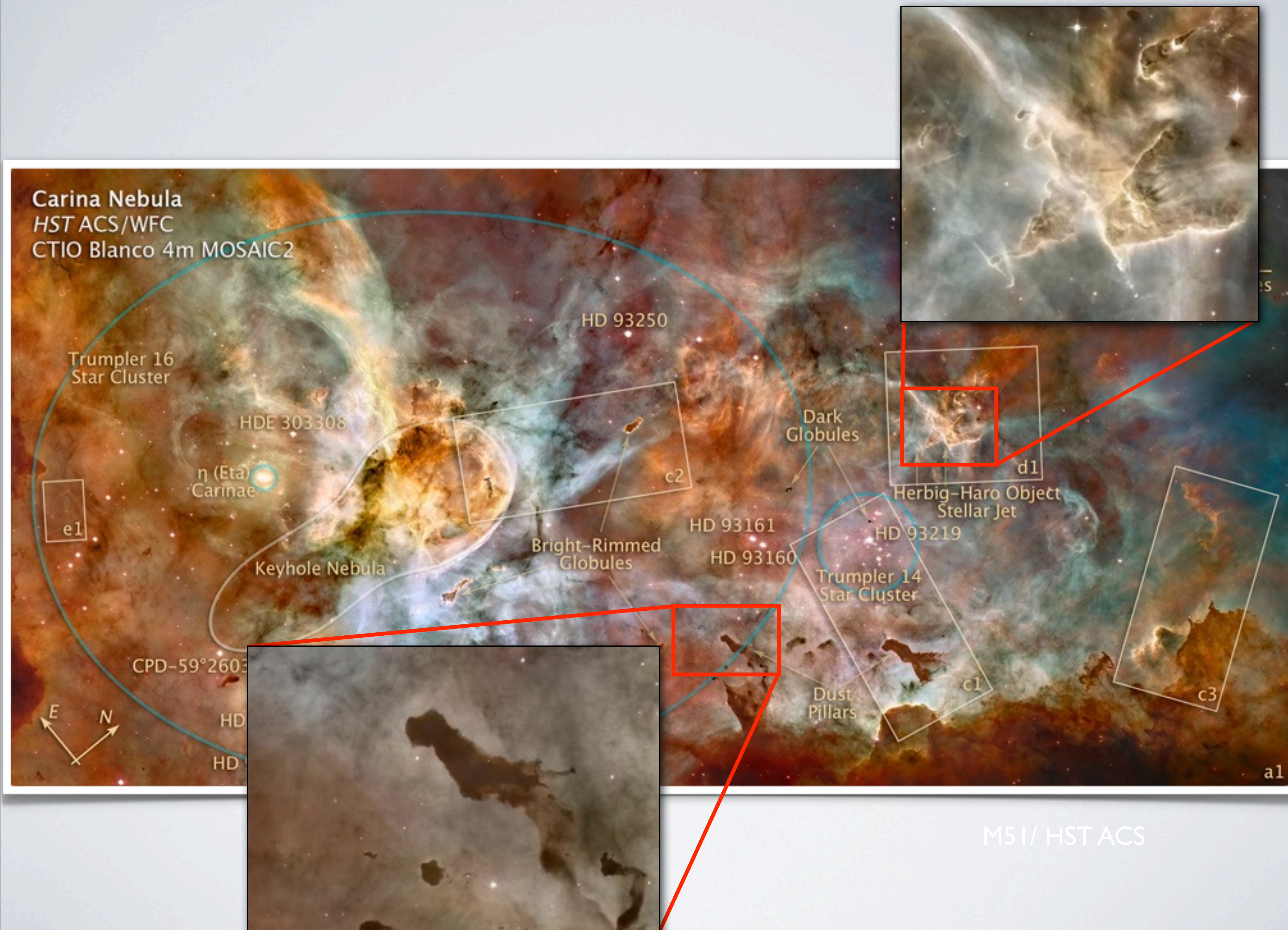
Star Formation = Gas→Star

Star Formation = Gas→Star



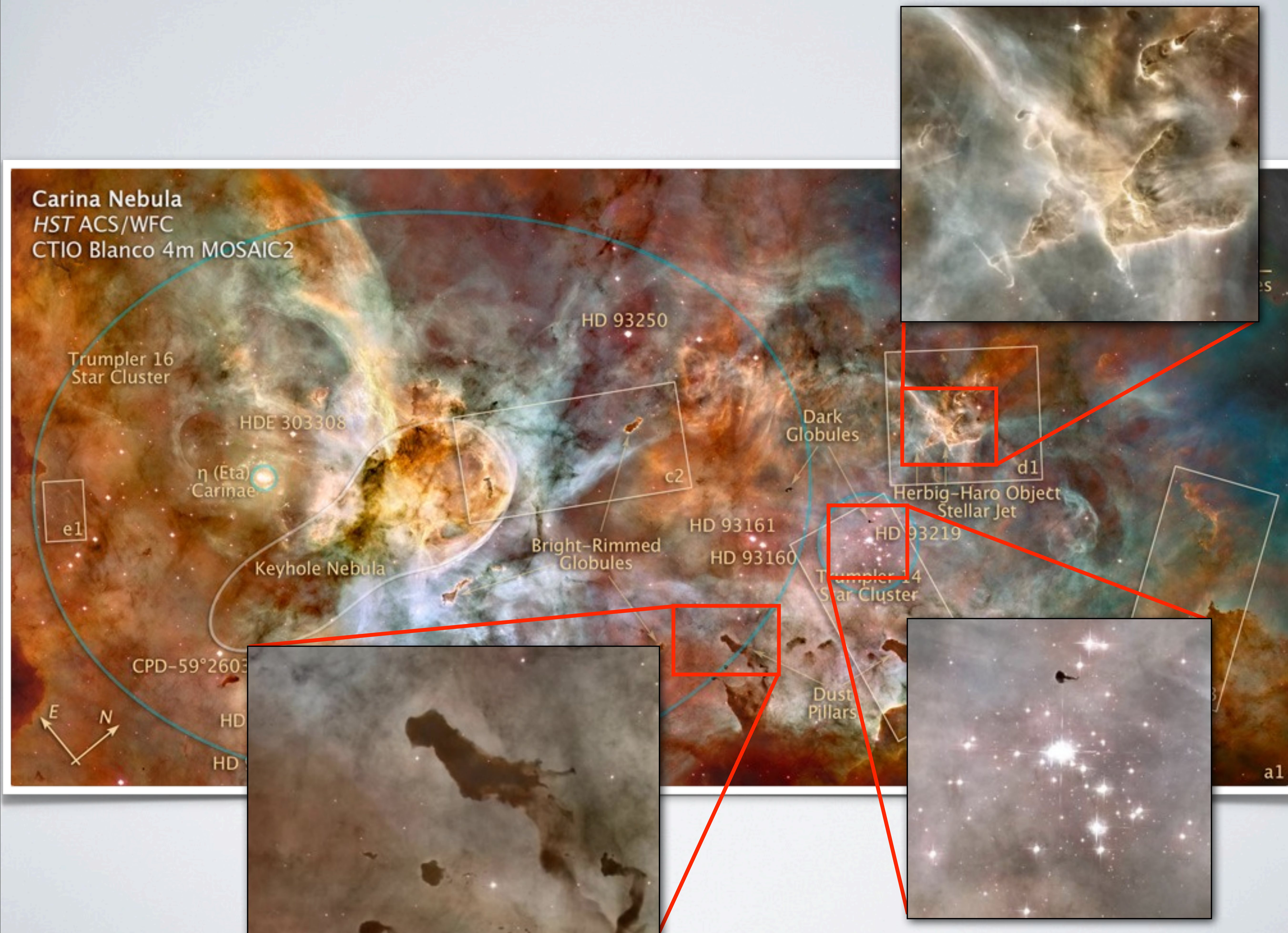
Star Formation = Gas→Star

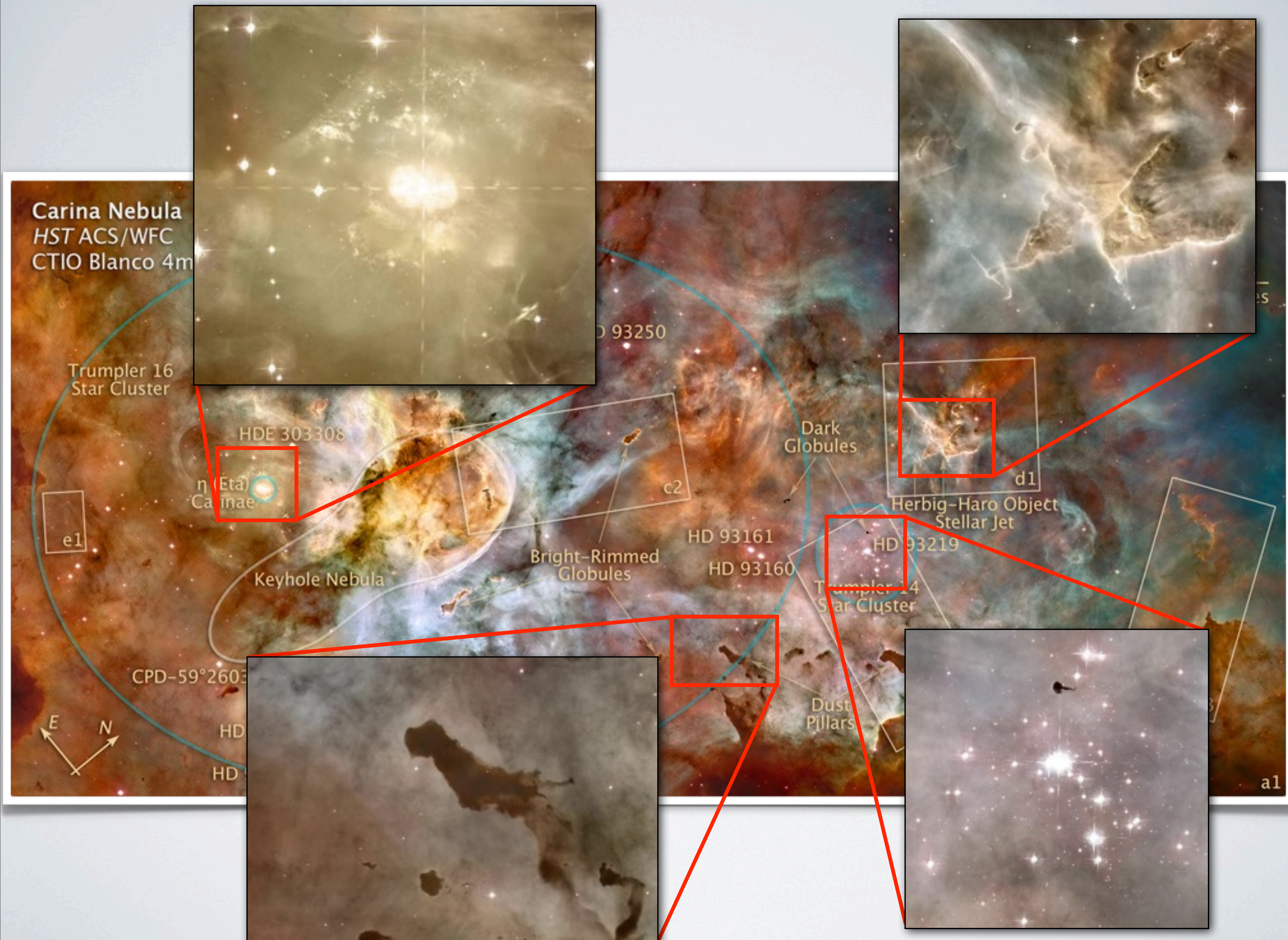
M51/ HST ACS



Carina Nebula
HST ACS/WFC
CTIO Blanco 4m MOSAIC2
Trumpler 16
Star Cluster
HDE 303308
η (Eta)
Carinae
e1
Keyhole Nebula
CPD-59°2603
E
N
HD 93250
c2
Bright-Rimmed
Globules
HD 93161
HD 93160
Dark
Globules
d1
Herbig-Haro Object
Stellar Jet
HD 93219
Trumpler 14
Star Cluster
Dust
Pillars
c1
c3
a1

M51/ HST ACS

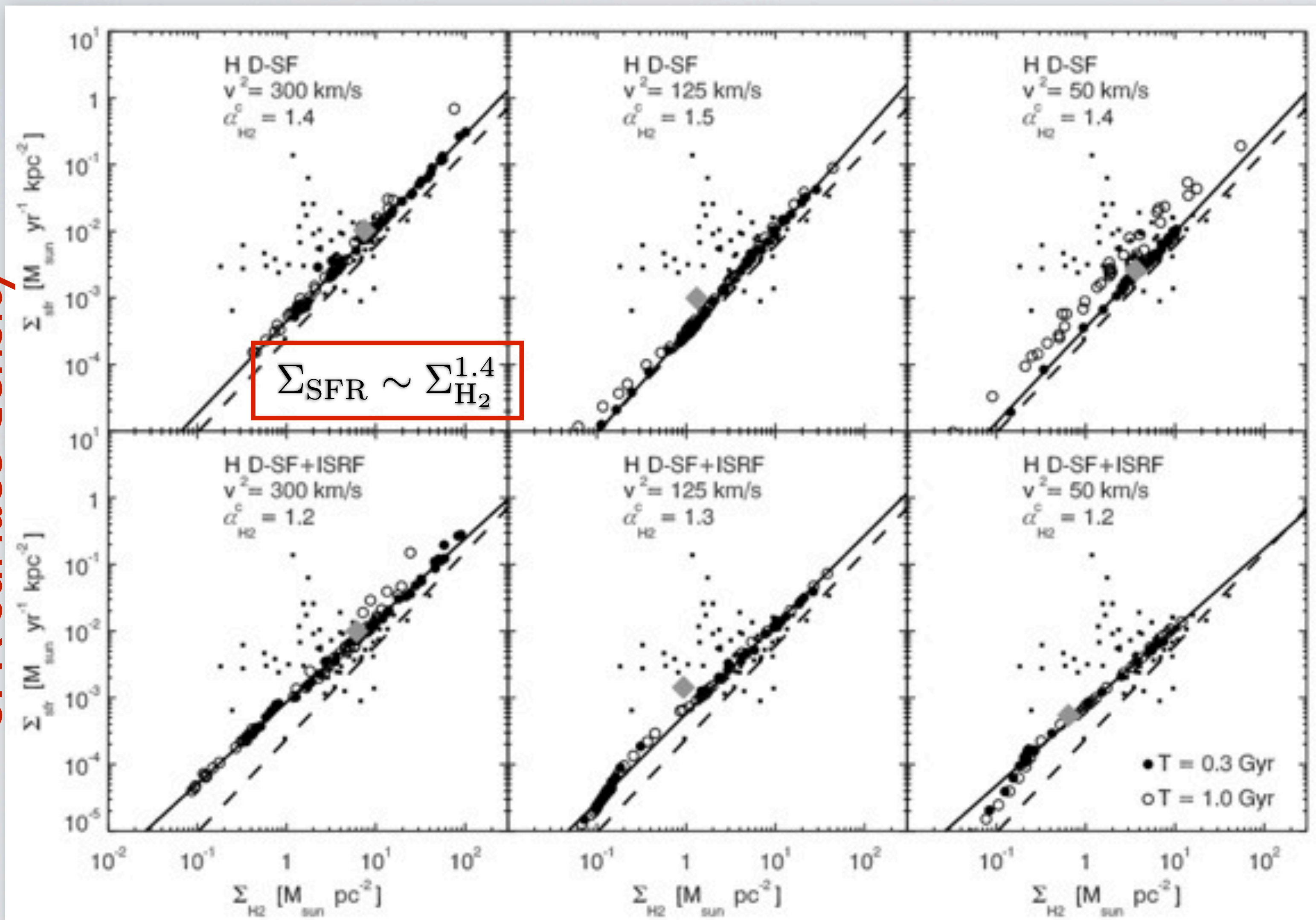

Carina Nebula
HST ACS/WFC
CTIO Blanco 4m MOSAIC2
HD 93250
Trumpler 16
Star Cluster
HDE 303308
η (Eta)
Carinae
e1
c2
Dark
Globules
d1
Herbig-Haro Object
Stellar Jet
HD 93161
HD 93219
HD 93160
Bright-Rimmed
Globules
Keyhole Nebula
Trumpler 14
Star Cluster
CPD-59°2603
Dust
Pillars
E
N
a1


Carina Nebula
HST ACS/WFC
CTIO Blanco 4m
Trumpler 16
Star Cluster
HDE 303308
η (Eta)
Carinae
e1
Keyhole Nebula
CPD-59°2603
E
N
HD
HD
93250
c2
Bright-Rimmed
Globules
Dark
Globules
HD 93161
HD 93160
d1
Herbig-Haro Object
Stellar Jet
HD 93219
Trumpler 14
Star Cluster
Dust
Pillars
a1

Previous SF Recipes

(so far in particle-based simulations)

SFR surface density →



H_2 surface density →

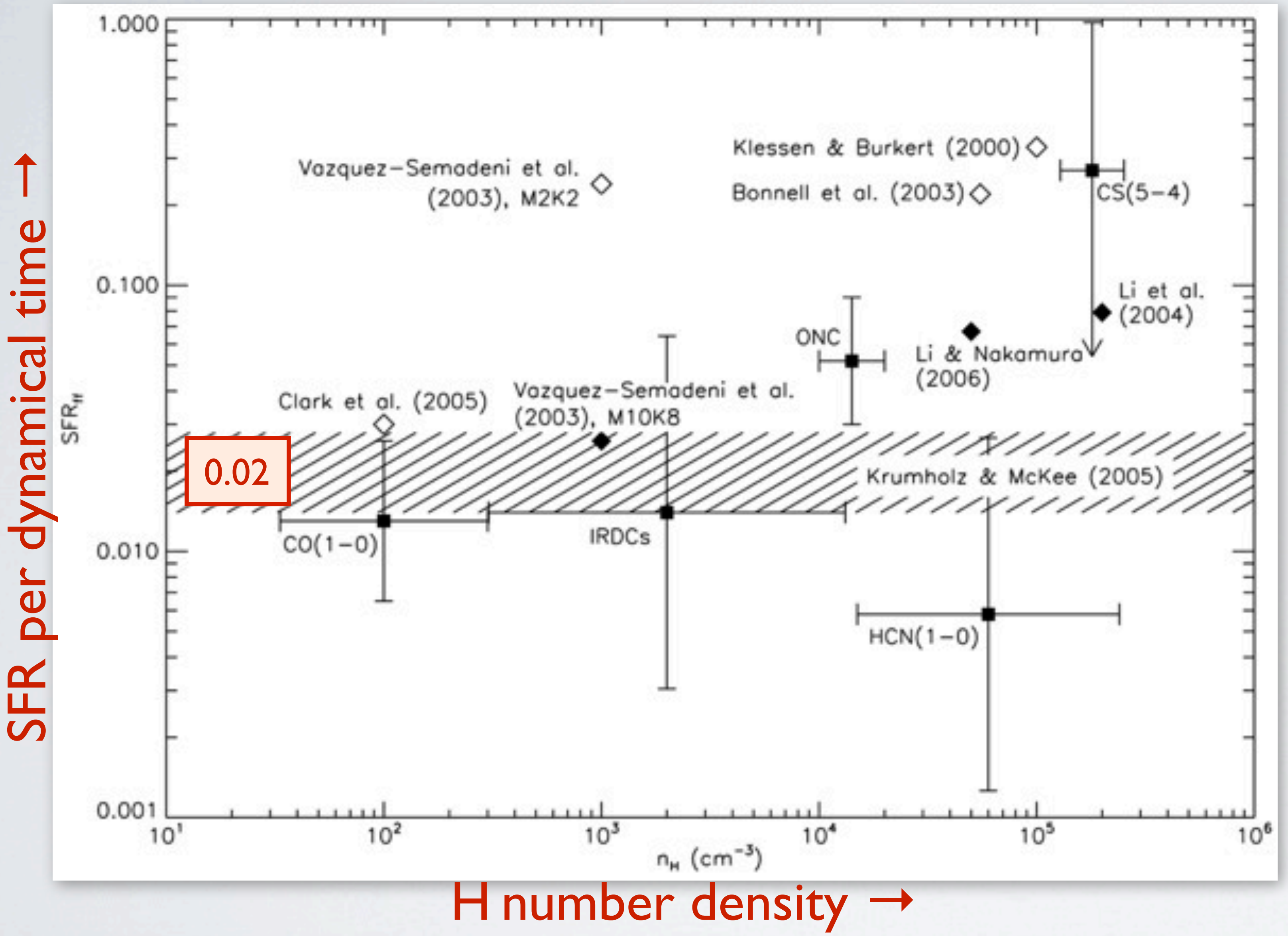
Robertson & Kravtsov (2008),
Kennicutt-Schmidt relation

- Dominated mostly by the SF recipe using the Schmidt relation (1959)

$$\dot{\rho}_* = (1-\beta) f_{\rm H_2} \frac{\rho_g}{t_*} \left(\frac{n_{\rm H}}{10 h^2 \ {\rm cm^{-3}}} \right)^{0.5}$$

- Apply thermal feedback or effective EOS to describe SNe feedback

Slow SF in Molecular Clouds



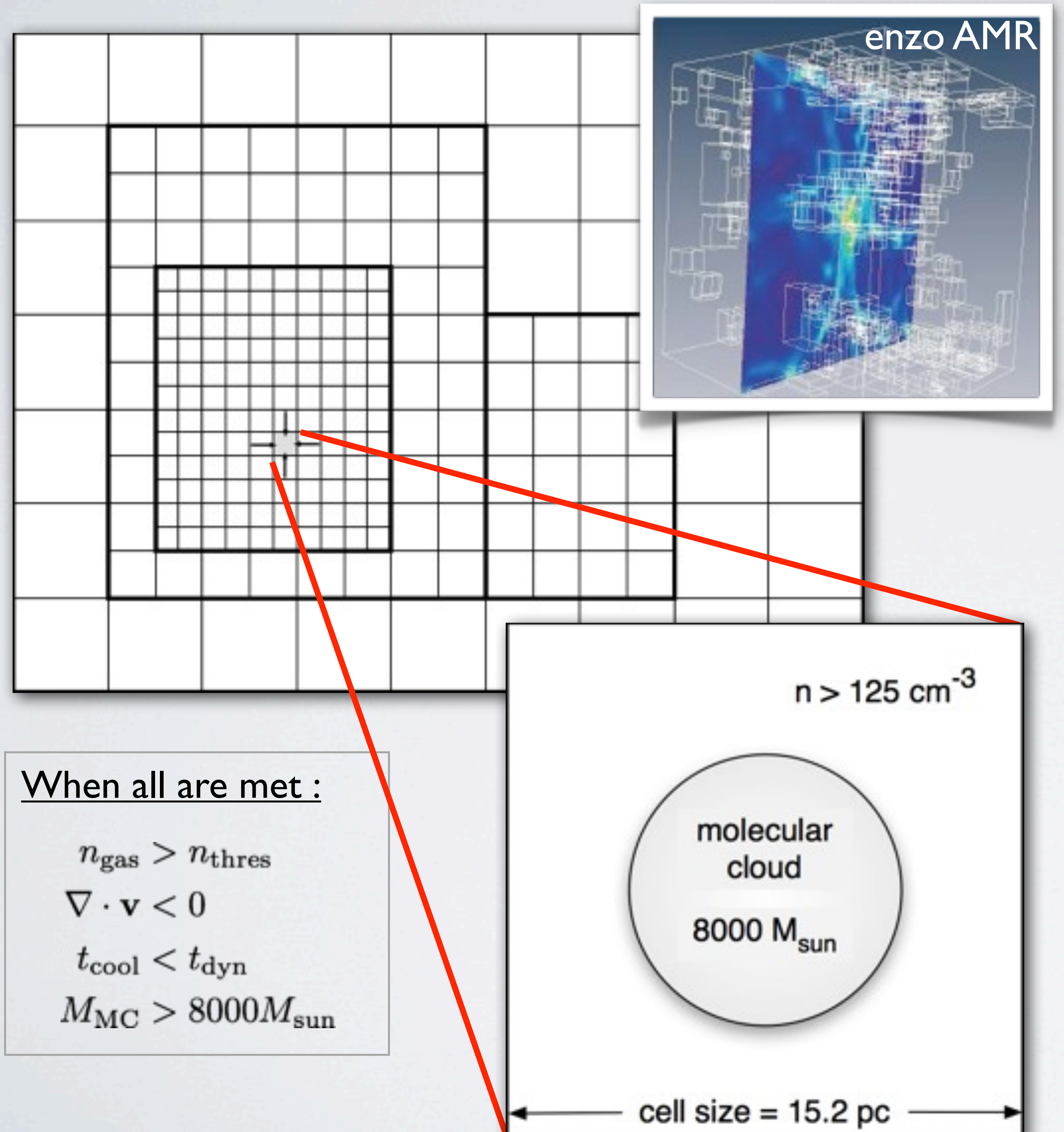
Krumholz & Tan (2007)

- Very slow due to turbulence, B-field, protostellar wind, etc.; should be reflected in galaxy-scale studies

$$SFR_{\rm ff} \sim 0.02$$

- MCs (10^4-10^5 M_{sun}) could be the basic units that can be represented in galaxy formation sims

MC Particle - Formation



When all are met :

$n_{\rm gas} > n_{\rm thres}$

$\nabla \cdot \mathbf{v} < 0$

$t_{\rm cool} < t_{\rm dyn}$

$M_{\rm MC} > 8000 M_{\rm sun}$

- Max resolution of 15.2 pc = $L_{\rm Jeans}$ of a MC of 125 particles/cm^3 at 960 K

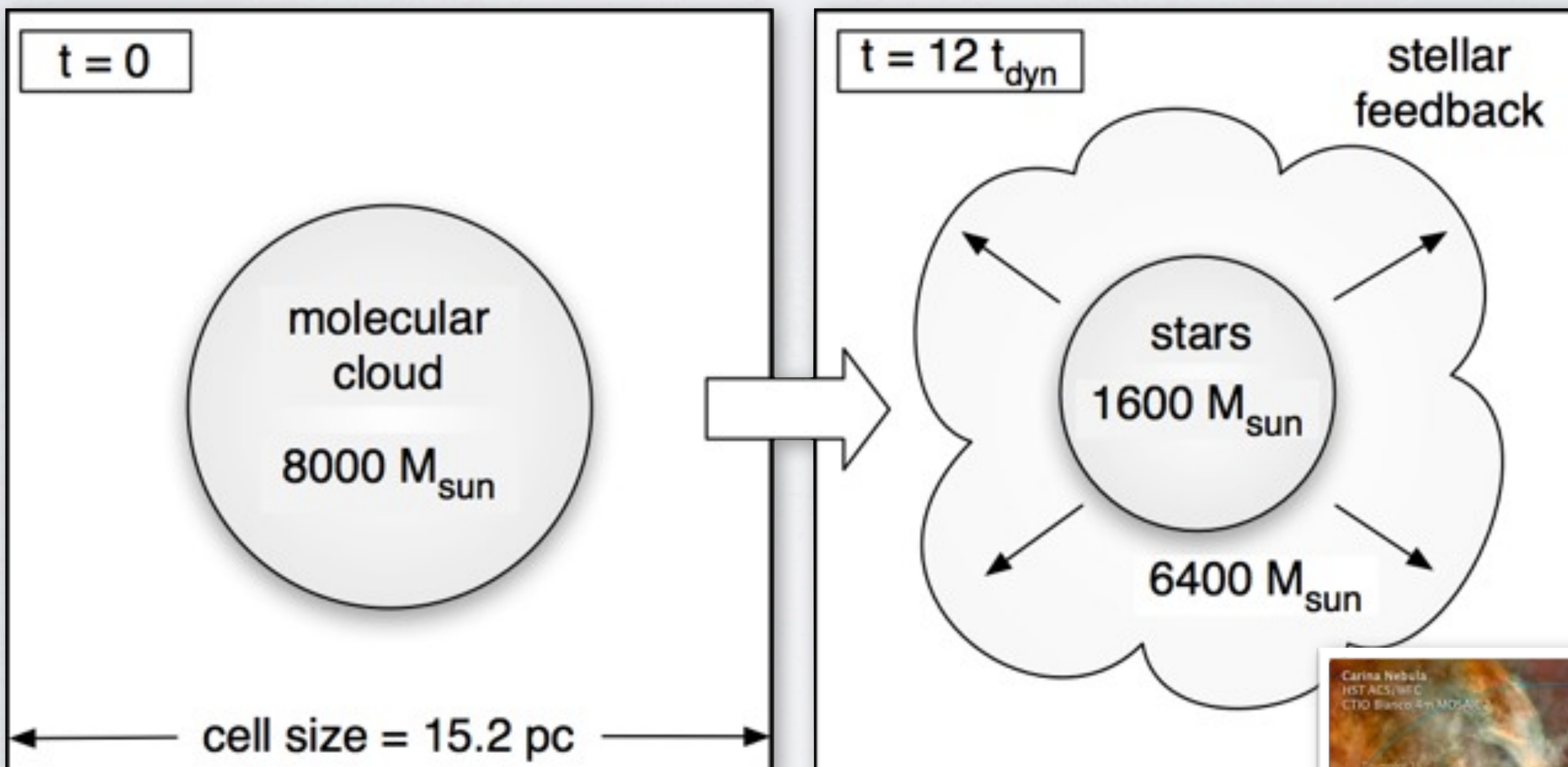
$$M_{\rm MC} = \epsilon_* \rho_{\rm gas} \Delta x^3$$

- Self-consistently deposit a particle when a cell of a typical MC size actually becomes Jeans unstable

  → each particle describes a MC of 8000 $M_{\rm sun}$

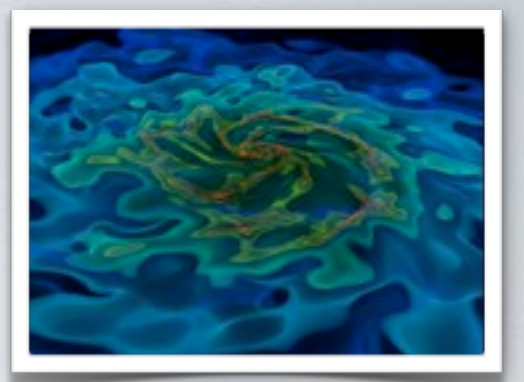
MC Particle - Feedback

- Both mass and energy are added back to gas
  - 80% of the MC mass slowly comes back to gas for 12 t_{dyn}
  - carries the thermal energy of 10^{51} ergs per M_{star}=750 M_{sun}



$$M_{\rm star}(\tau) = 0.2\, M_{\rm MC} \int_0^{\tau} \tau' e^{-\tau'}\, d\tau'$$

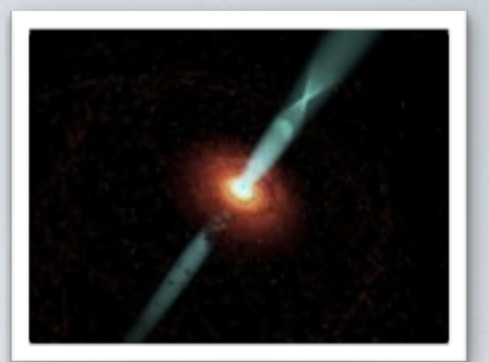
PART I



[Star Formation and Feedback]

MC

PART II

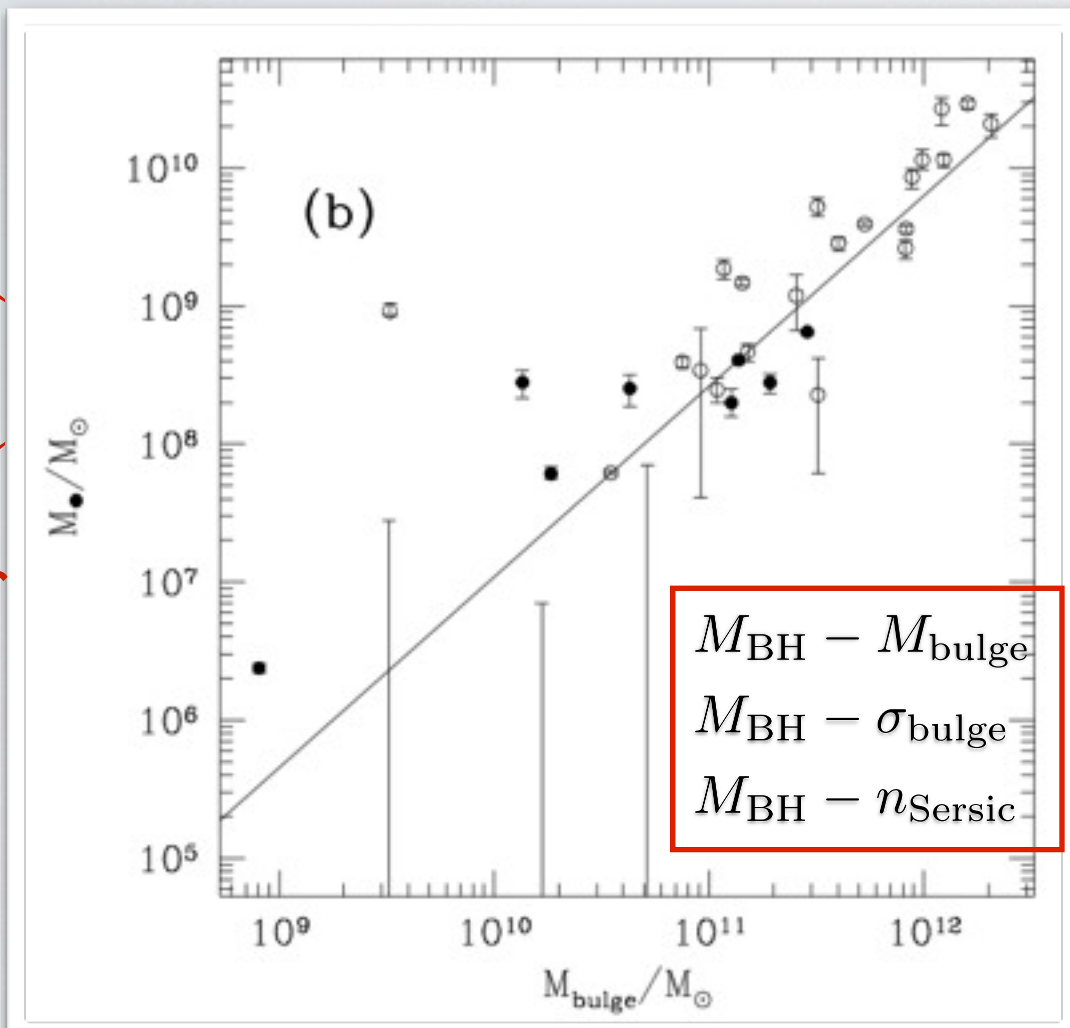


[MBH Accretion and Feedback]

Coevolution of Galaxies and MBHs

- Have galaxies and MBHs grown at the same time under each other's influence?

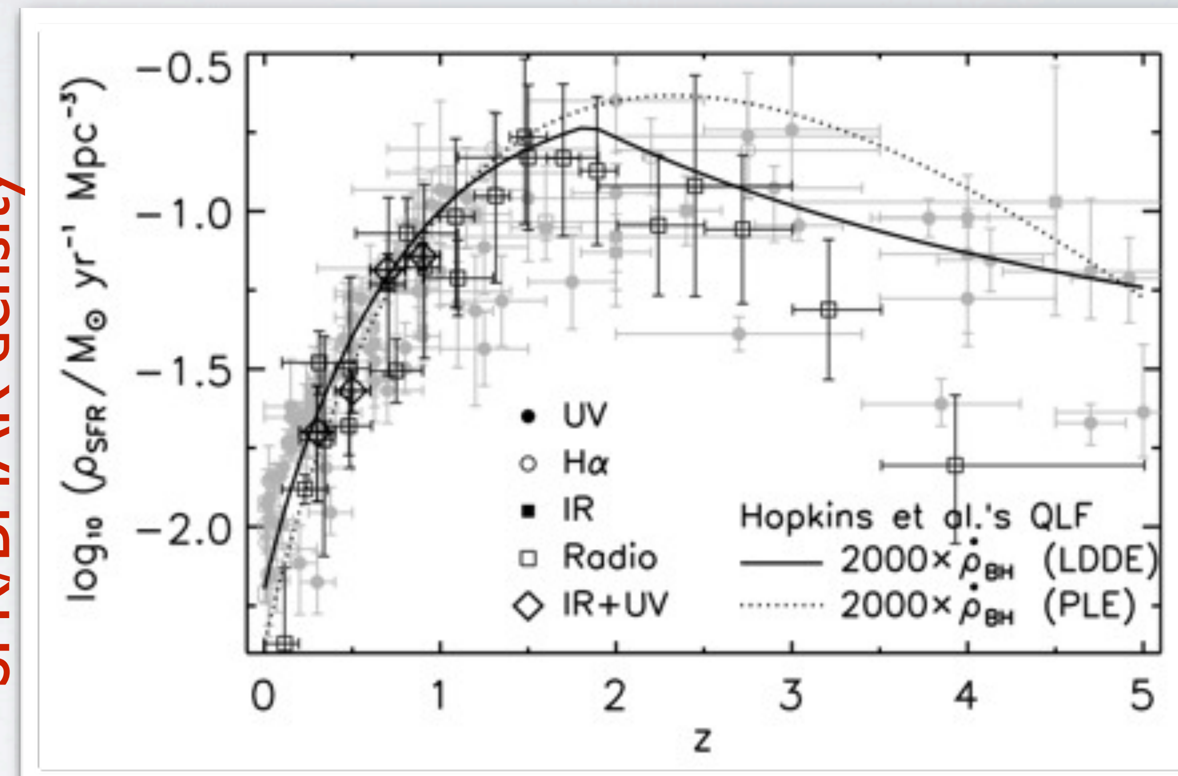
massive dark object (MDO) mass →



bulge mass →

Magorrian et al. (1998)

SFR/BHAR density →



redshift →

Zheng et al. (2009)

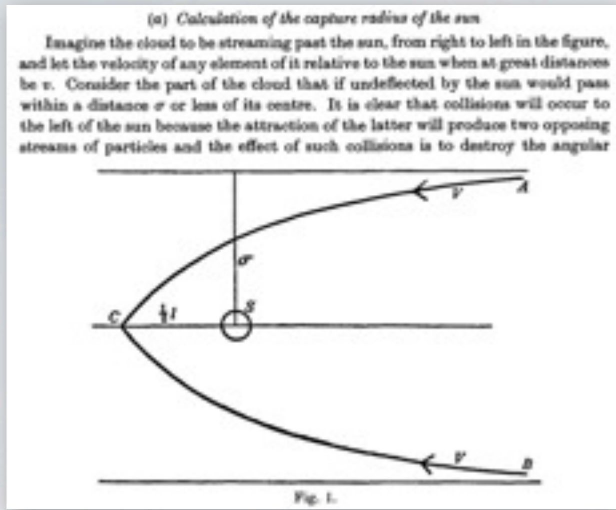
* Unified model: Silk & Rees (1998), Kauffmann & Haehnelt (2000), etc.

Second Component!

- GOAL: Study the coevolution of galaxies and MBHs in one comprehensive self-consistent framework!

Previous Sink Particle Recipe

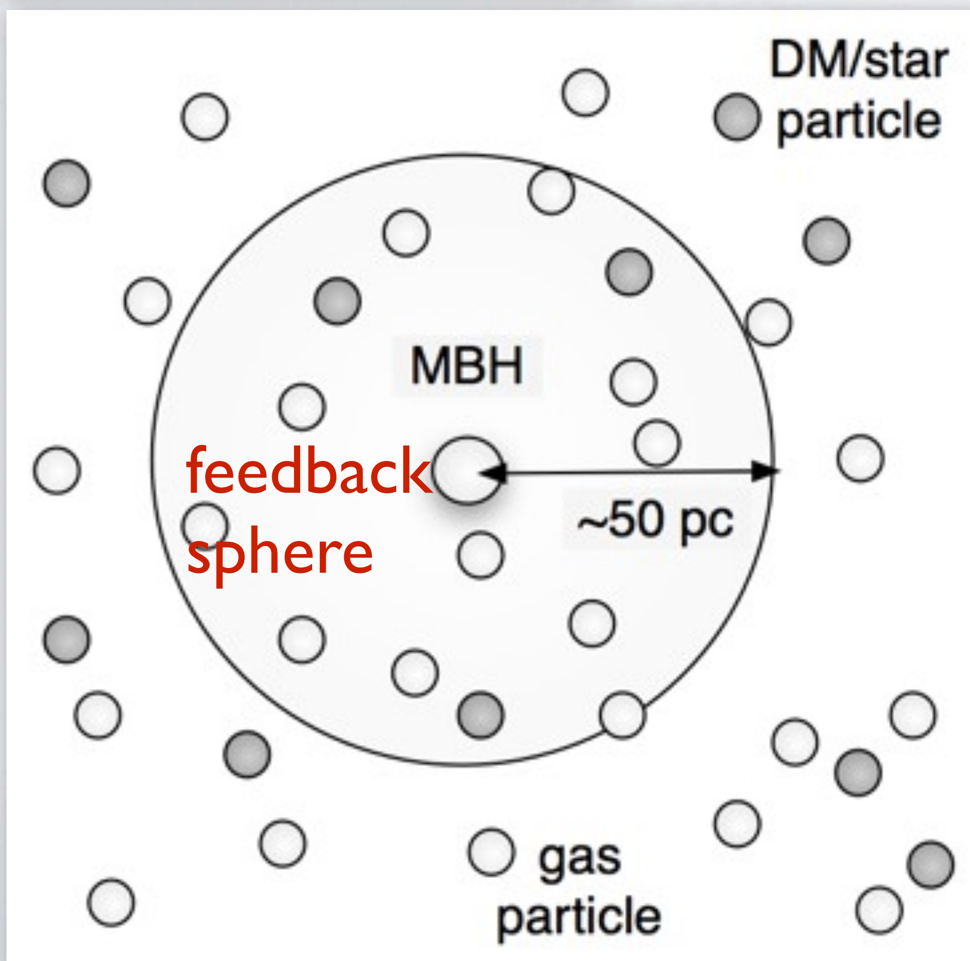
(so far in particle-based galaxy formation)


(a) Calculation of the capture radius of the sun

Imagine the cloud to be streaming past the sun, from right to left in the figure, and let the velocity of any element of it relative to the sun when at great distances be v. Consider the part of the cloud that if undeflected by the sun would pass within a distance σ or less of its centre. It is clear that collisions will occur to the left of the sun because the attraction of the latter will produce two opposing streams of particles and the effect of such collisions is to destroy the angular

Fig. 1.

Hoyle & Lyttleton (1939)



Springel et al. (2005) & many others

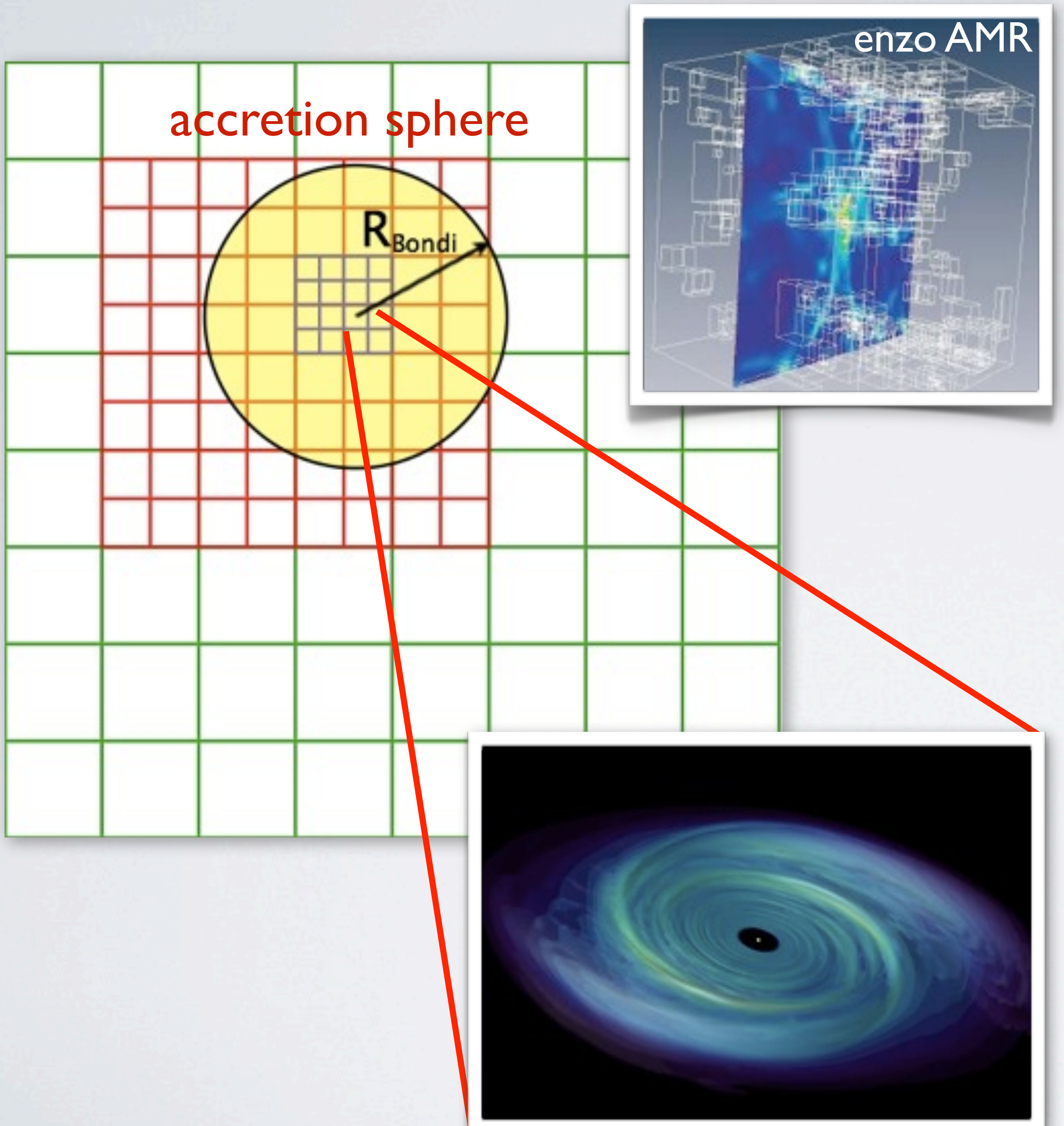
- Growing MBH based on the spherical Bondi-Hoyle accretion argument

$$\dot{M}_{\rm BH} = \texttt{min}\left(\frac{4\pi\alpha G^2 M_{\rm BH}^2 \rho_{\rm B}}{c_{\rm s}^3}, \frac{4\pi G M_{\rm BH} m_{\rm p}}{\epsilon_{\rm r}\sigma_{\rm T} c}\right)$$

- Kernel-weighted thermal feedback (5% most cases) based on accretion rate

$$\dot{E}_{\rm BH,th} = \epsilon_{\rm f}\epsilon_{\rm r}\dot{M}_{\rm BH}c^2$$

MBH Particle - Accretion



- Eddington-limited Bondi estimate with no tweaks; subtraction from a sphere of radius $R_{\rm Bondi}$

$$\dot{M}_{\rm BH} = \min\left(\frac{4\pi G^2 M_{\rm BH}^2 \rho_{\rm B}}{c_{\rm s}^3}, \frac{4\pi G M_{\rm BH} m_{\rm p}}{\epsilon_{\rm r} \sigma_{\rm T} c}\right)$$

- Getting close to resolving $R_{\rm Bondi}$ of MBHs in galaxy-scale simulations

$$R_{\rm Bondi} = \frac{2GM_{\rm BH}}{c_{\rm s}^2} \simeq 8.6 \ {\rm pc} \left(\frac{M_{\rm BH}}{10^5 M_\odot}\right)\left(\frac{10\,{\rm km/s}}{c_{\rm s}}\right)^2$$

MBH Particle - Feedback

- Designed three different feedback channels; two currently in use

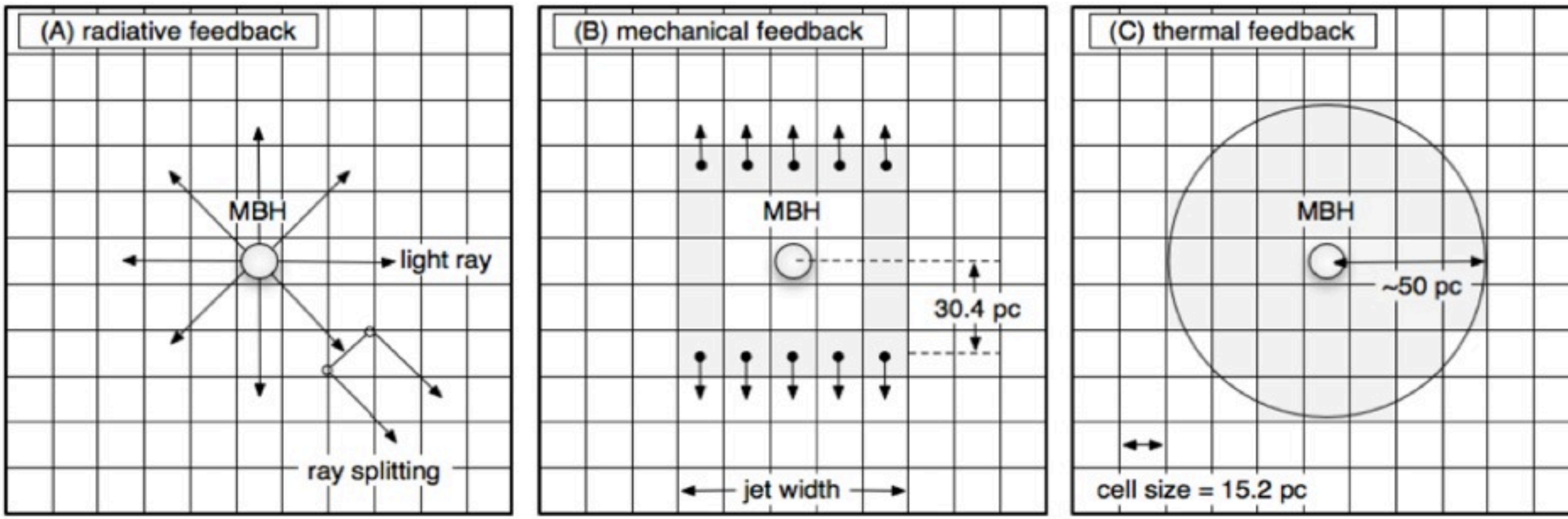
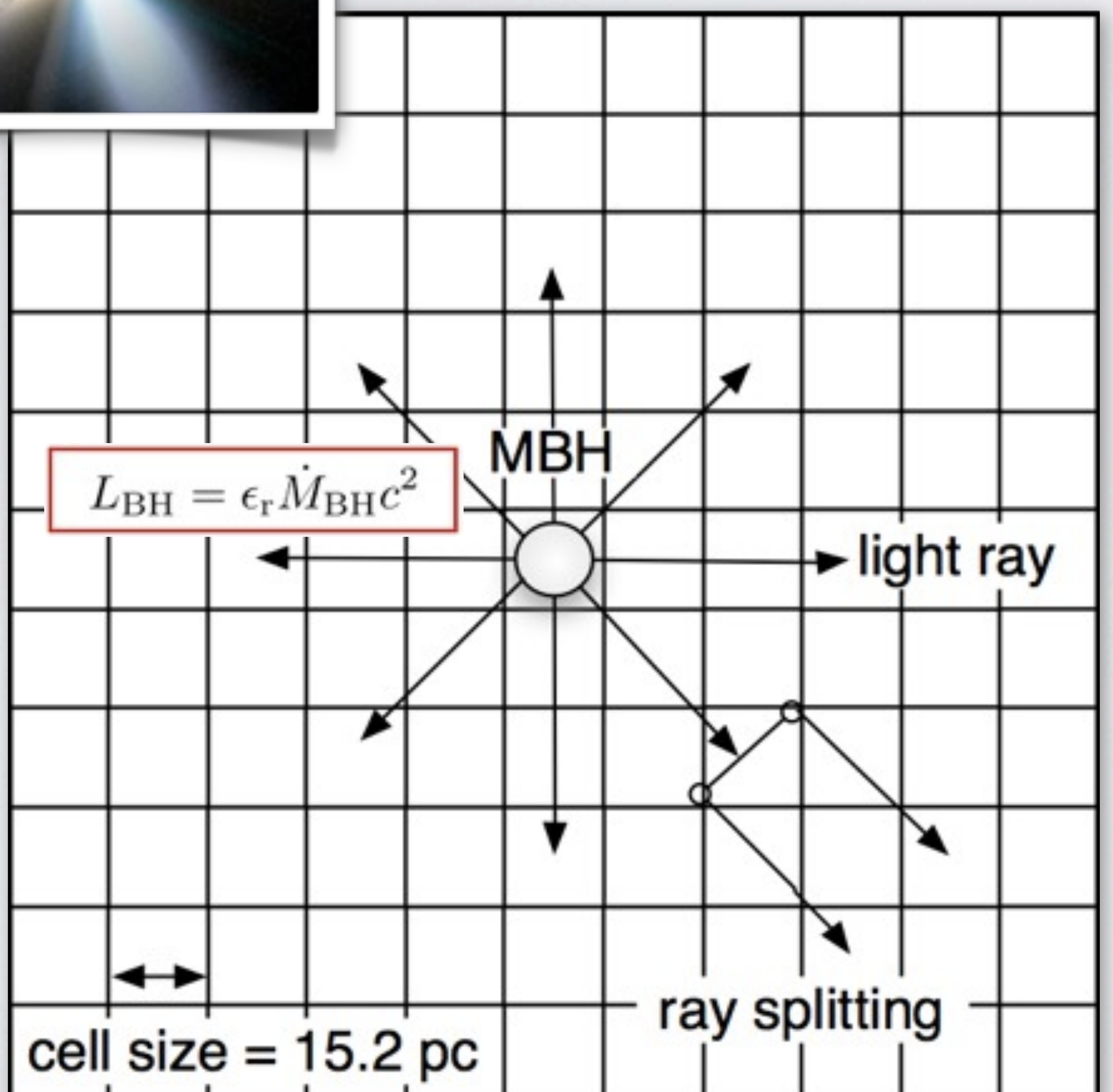


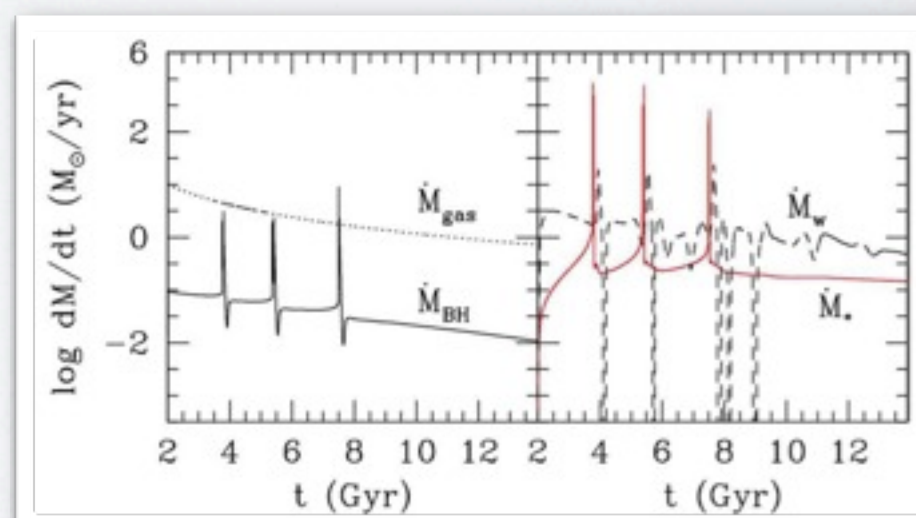
FIG. 2.— Two-dimensional schematic views of the different modes of massive black hole feedback. (A) radiative feedback model described in Section 2.7: photon packages carrying the energy is adaptively traced via full radiative transfer, (B) mechanical feedback model described in Section 2.8: a momentum is given to the cells around the MBH along a pre-calculated jet direction, and (C) thermal feedback model dominantly used in particle-based galaxy-scale simulations: energy is thermally deposited kernel-weighted to the neighboring gas particles around the MBH.

- Kim, Wise, Alvarez, & Abel (2010) in prep.

(1) MBH Radiative Feedback

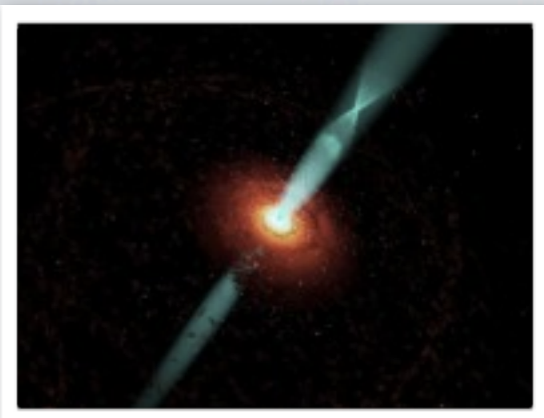


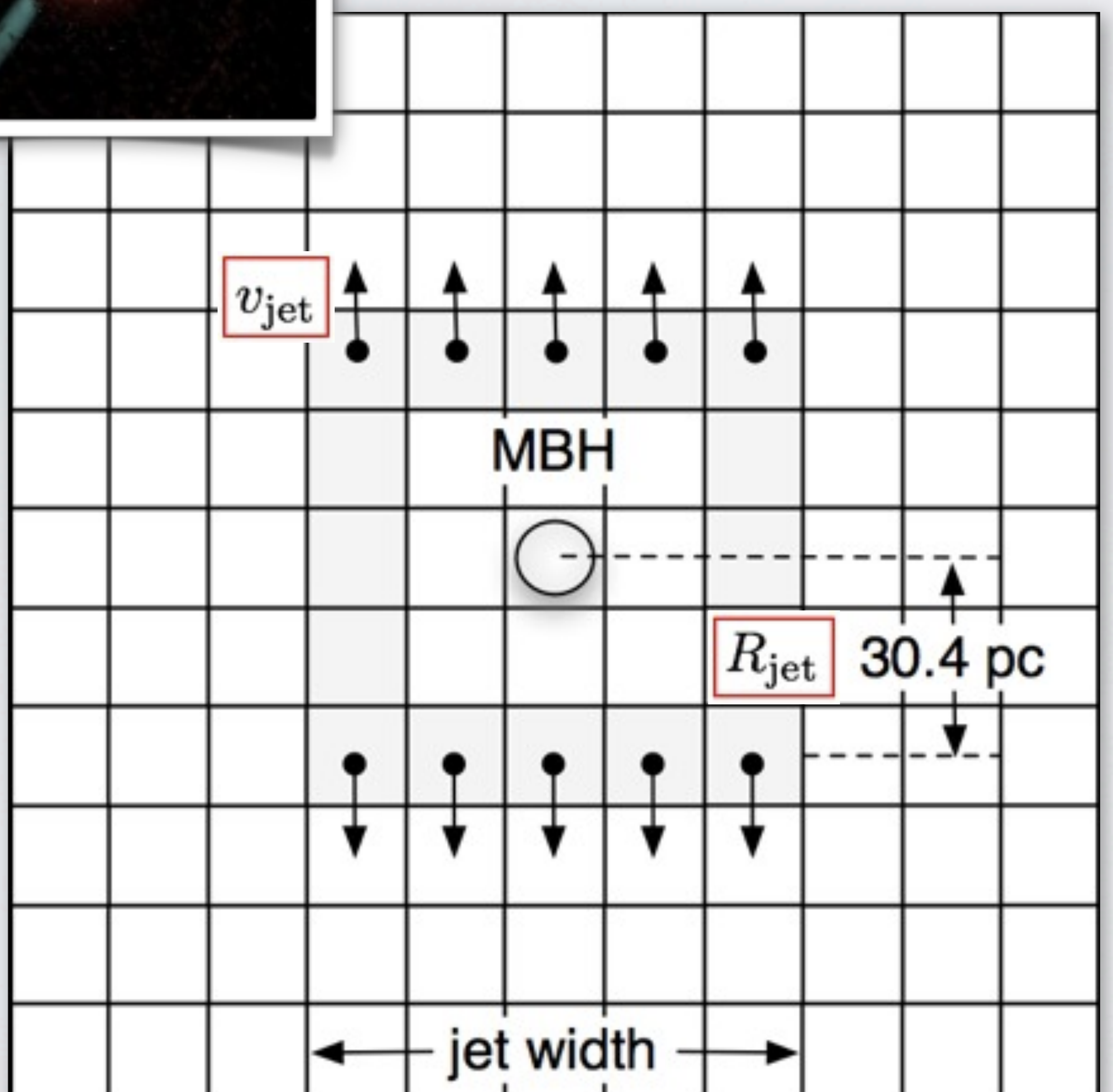
- Full 3D radiative transfer: monochromatic 2 keV X-ray photon packages do
  - photoionization (H,He,He^+)
  - photoheating
  - Compton heating (e^-)
  - radiation pressure



Ciotti et al. (2010): 1D-model

(2) MBH Mechanical Feedback





- Mechanical Energy
  = Potential Energy
  (jets introduced at R_{jet})
  + Kinetic Energy
  (jets launched with v_{jet})

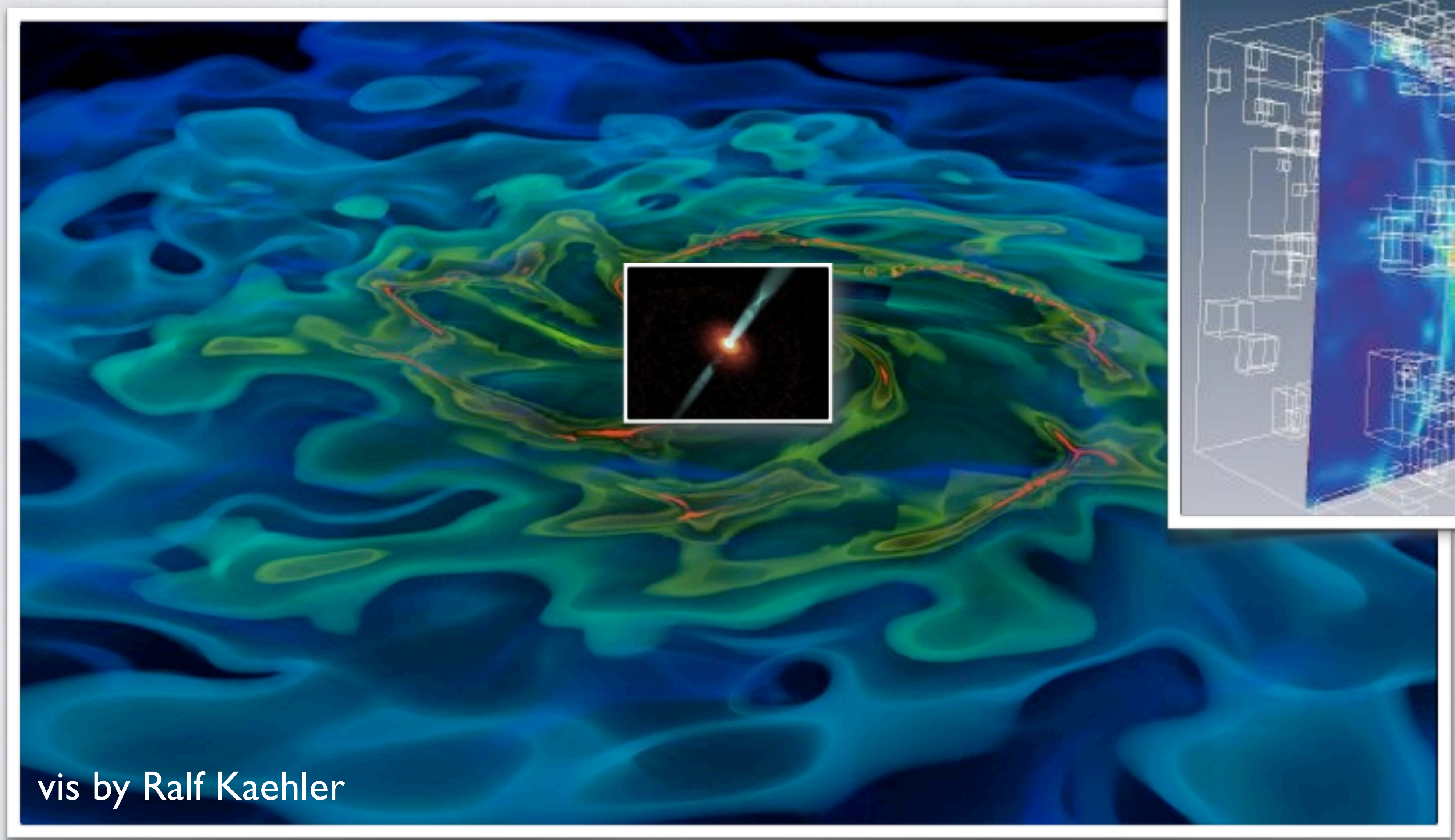
$$\epsilon_{kin} \equiv \frac{P_{kin}}{L_{BH}} = 10^{-4} \quad \text{and} \quad \eta_{jet} \equiv \frac{\dot{M}_{jet}}{\dot{M}_{BH}} = 0.05$$

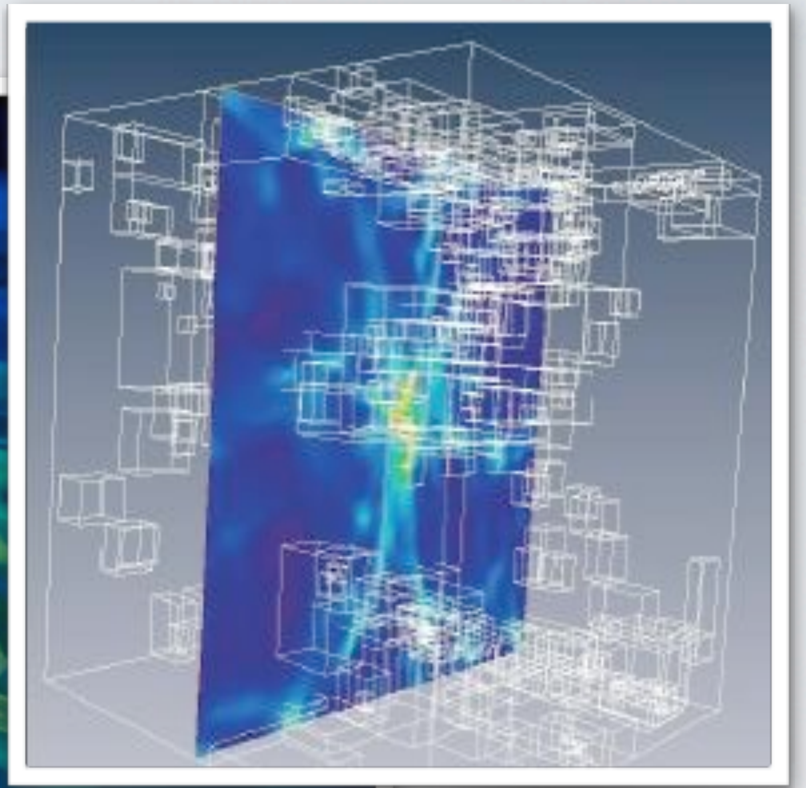
$$\rightarrow \quad v_{jet} = c\left(\frac{2\epsilon_{kin}\epsilon_r}{\eta_{jet}}\right)^{1/2}$$

- Directed along $\vec{L}_{gas\text{-}accreted}$; injected at every 300 M_{sun}

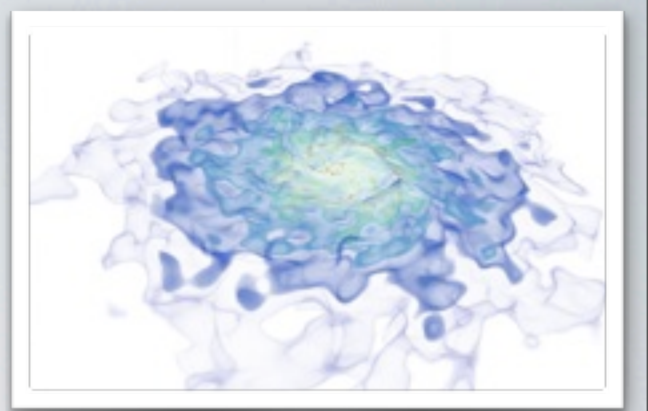
Multi-scale Physics

- Resolving things from R_{Bondi} to R_{galaxy}, from 10^2 K to 10^7 K
  → AMR enzo-2.0 poised to do a better job than ever





PART III



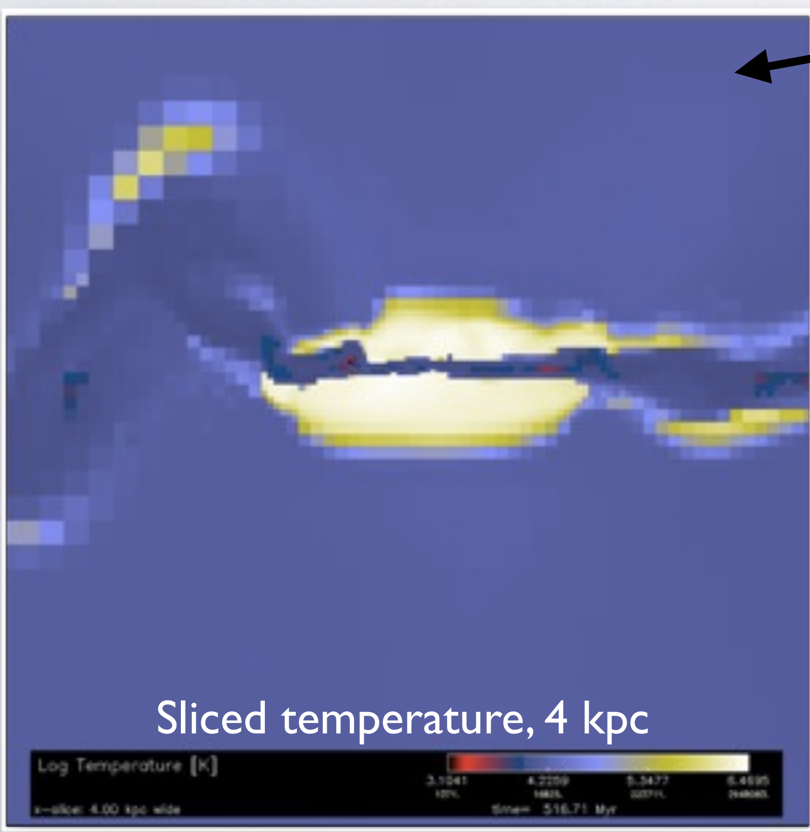
[Setting Up An Experiment & Early Results]

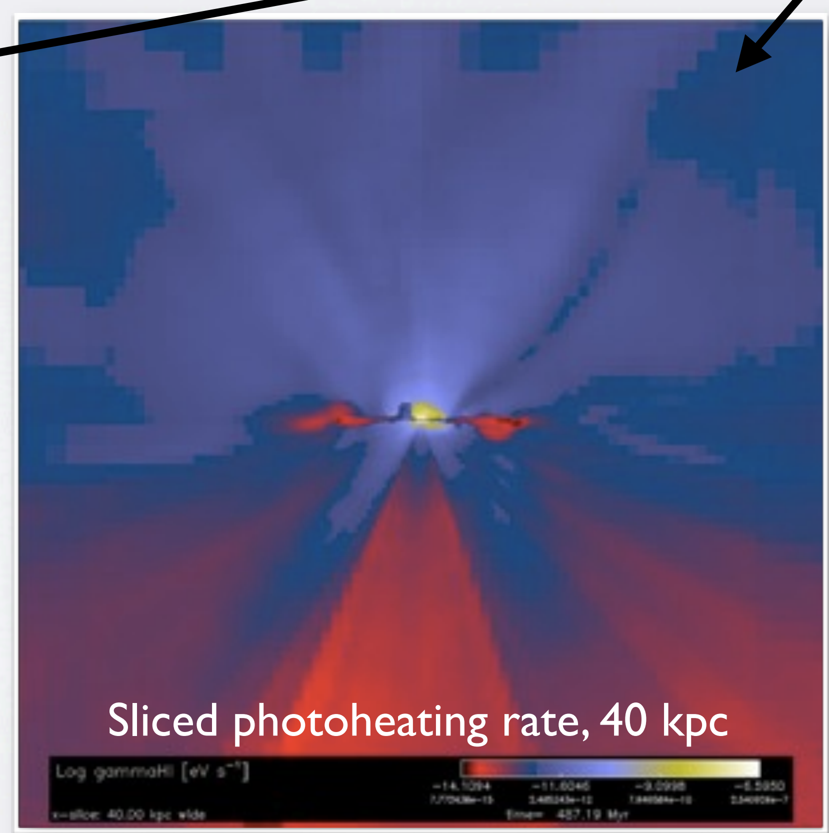
Simulation Suite

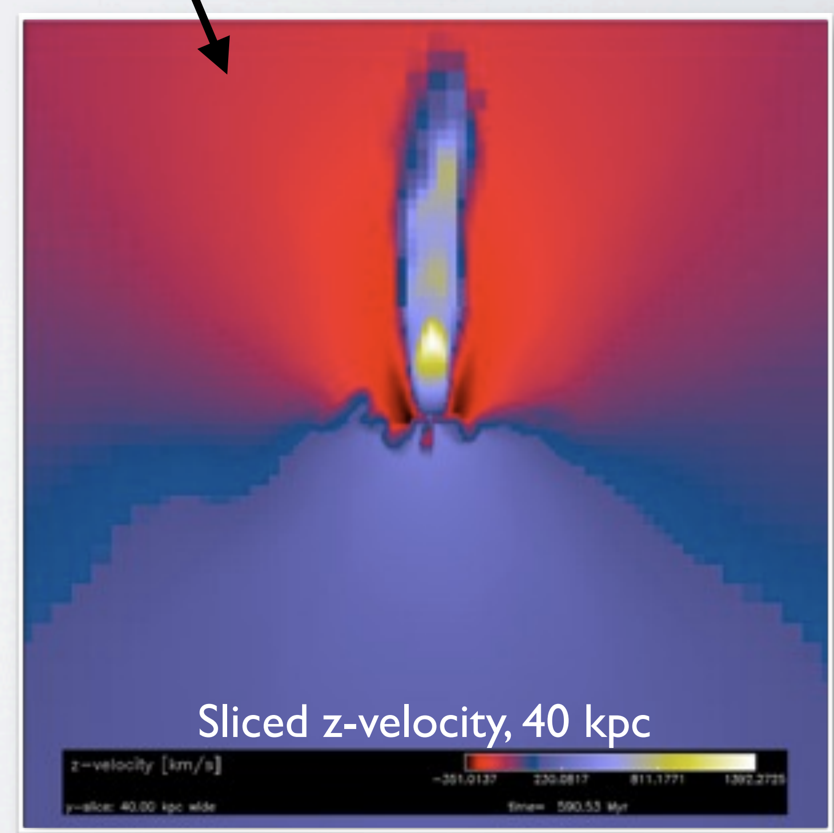
- Ensembles of simulations with different modes of feedback to study the galaxy evolution regulated by stellar and MBH feedback

TABLE 1
SIMULATION SUITE DESCRIPTION

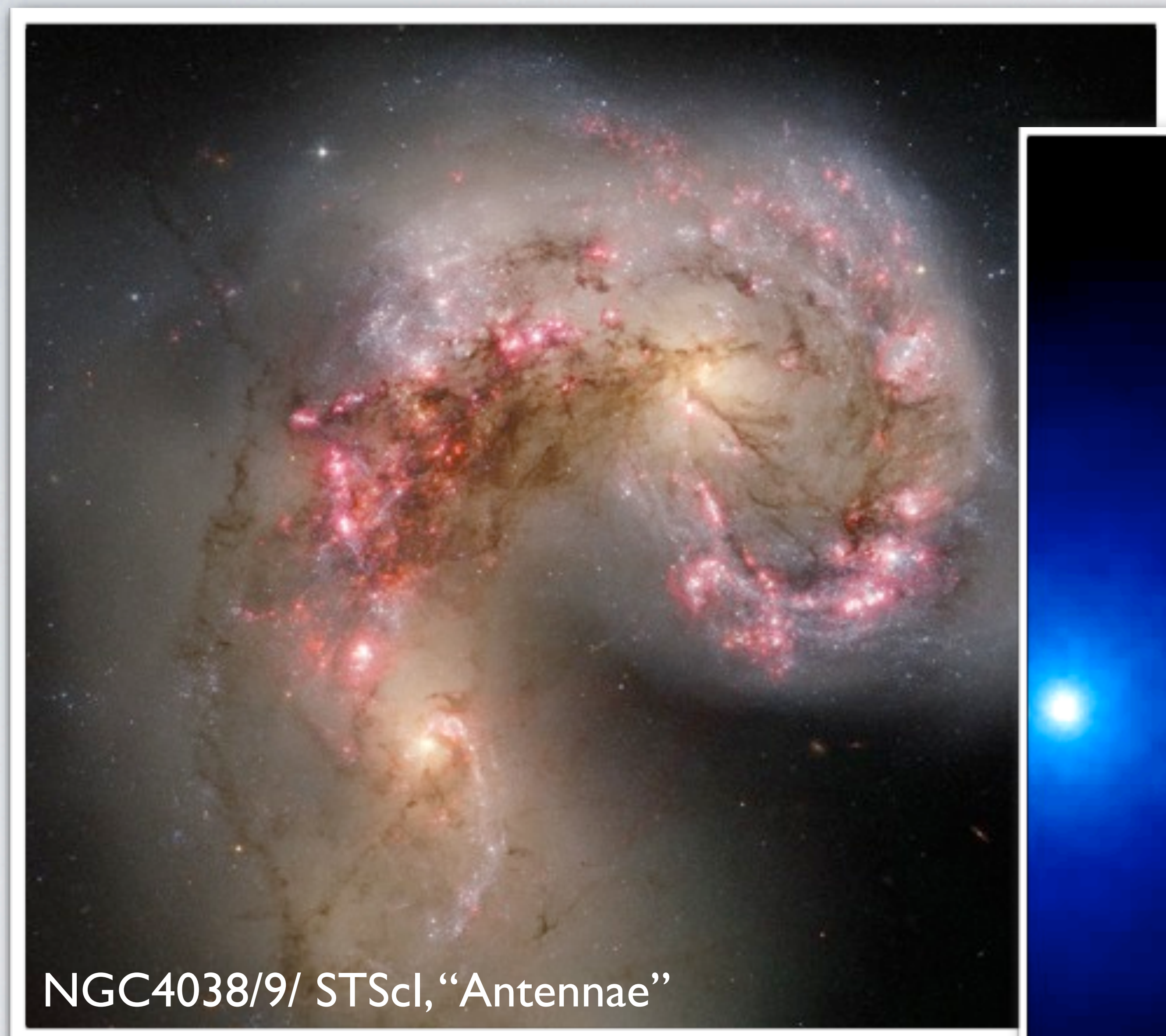
Physics[a]		Sim-NF	Sim-SF	Sim-RF	Sim-MF	Sim-RMF
Molecular cloud formation	(Section 2.4)	○	○	○	○	○
Stellar feedback	(Section 2.5)	×	○	○	○	○
Massive black hole accretion	(Section 2.6)	○	○	○	○	○
Massive black hole radiative feedback	(Section 2.7)	×	×	○	×	○
Massive black hole mechanical feedback	(Section 2.8)	×	×	×	○	○

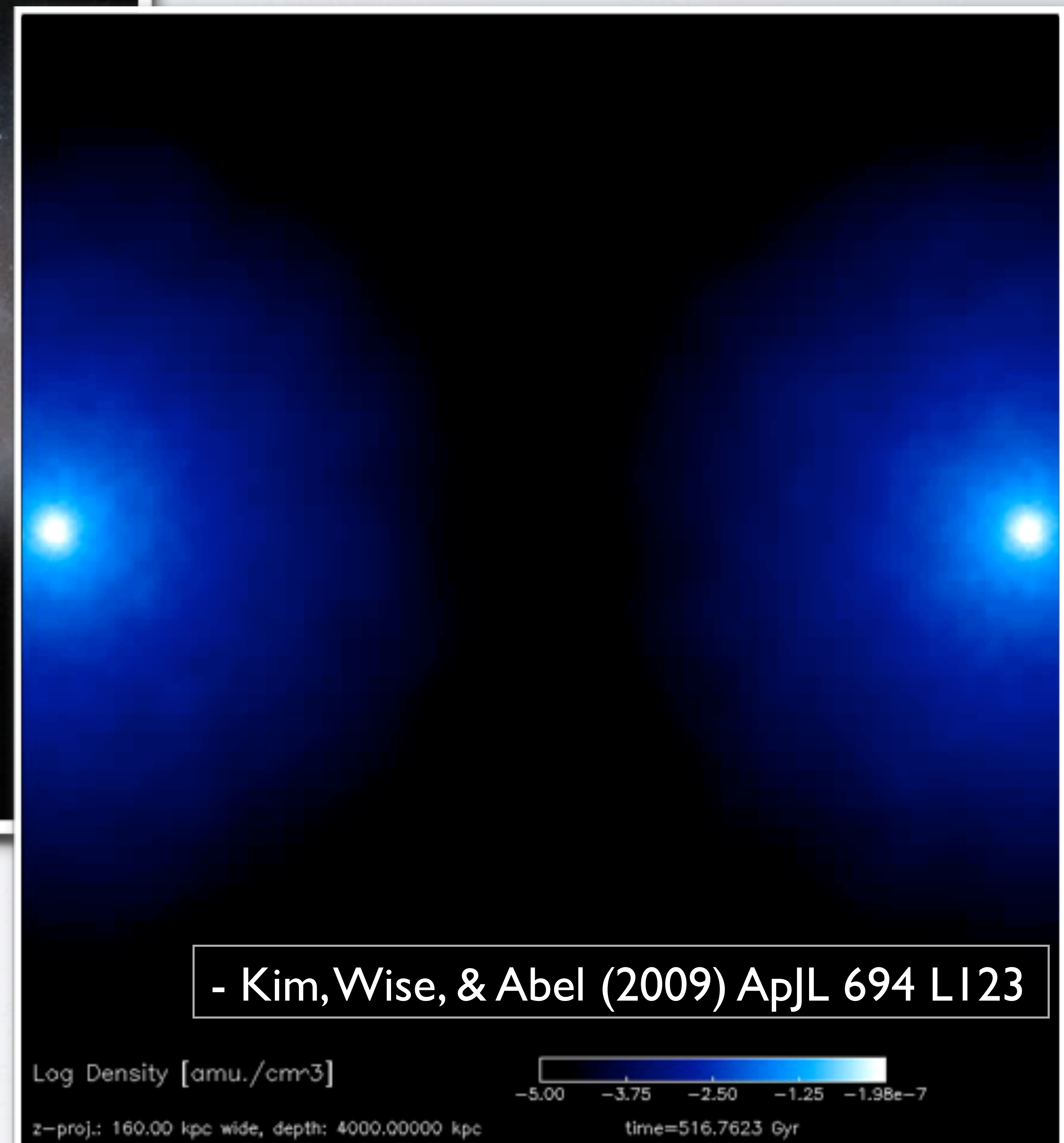

Sliced temperature, 4 kpc


Sliced photoheating rate, 40 kpc


Sliced z-velocity, 40 kpc

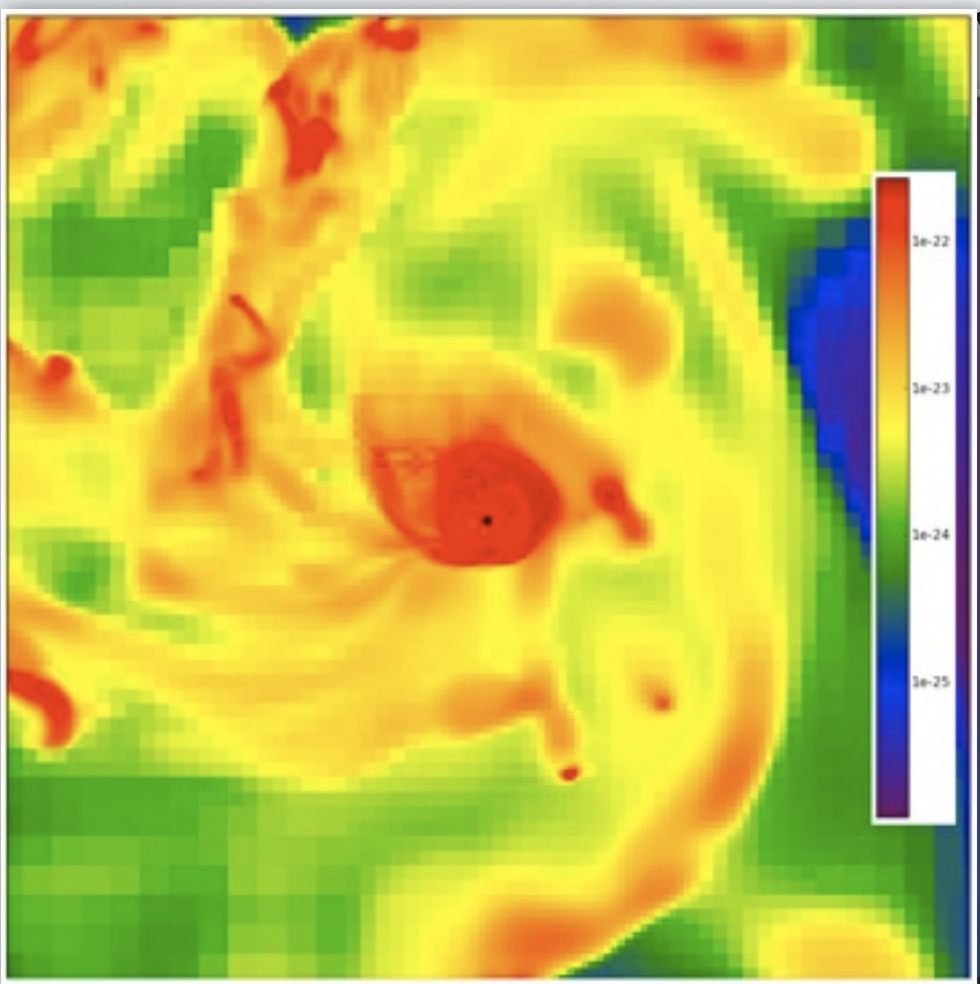
Galaxy Mergers: Great Laboratory



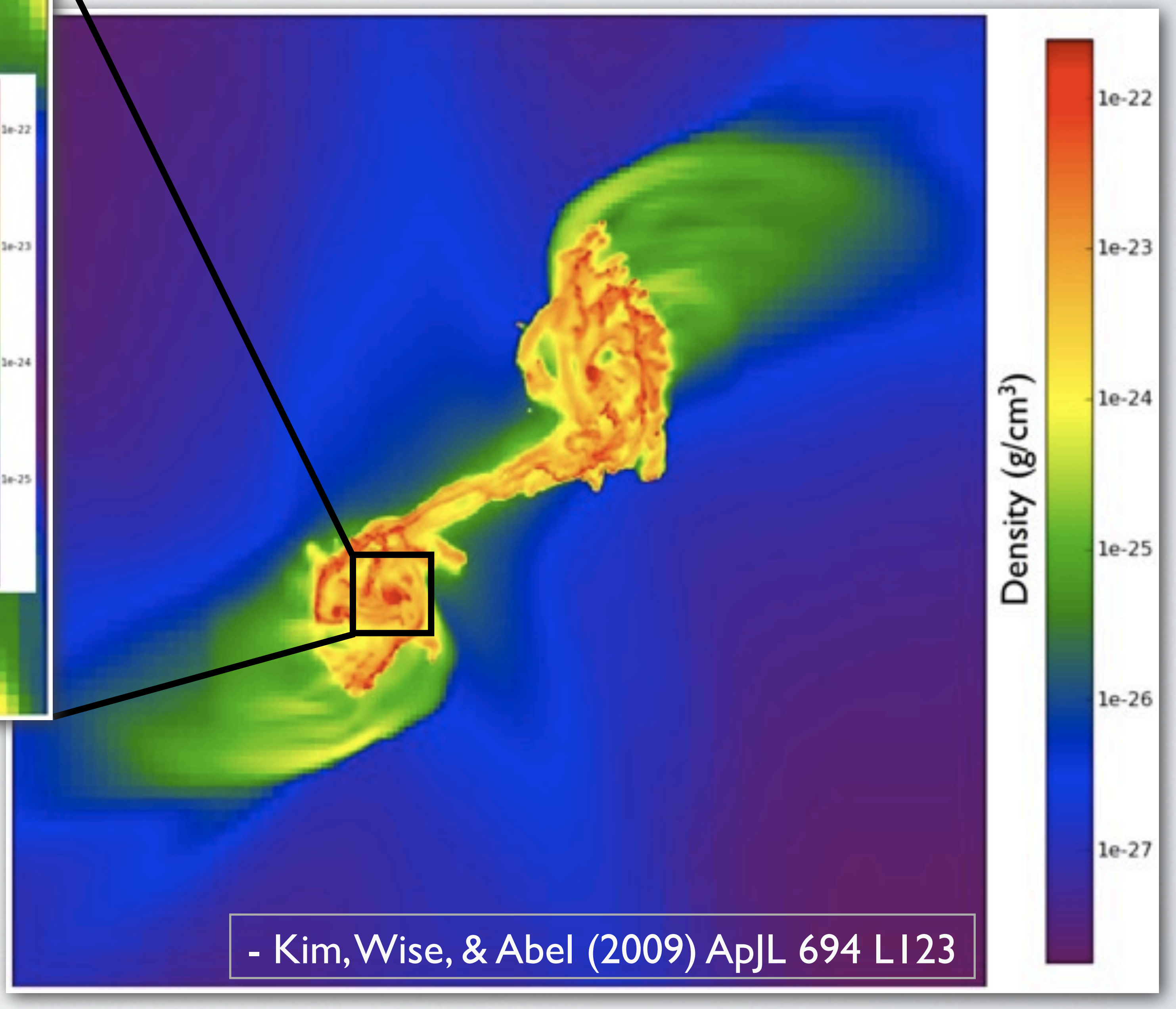


Galaxy Mergers

- Two 2×10^{11} M_{sun} galaxies with embedded 10^5 M_{sun} MBHs set on a collisional orbit (30° tilted, initially separated by 80 kpc)

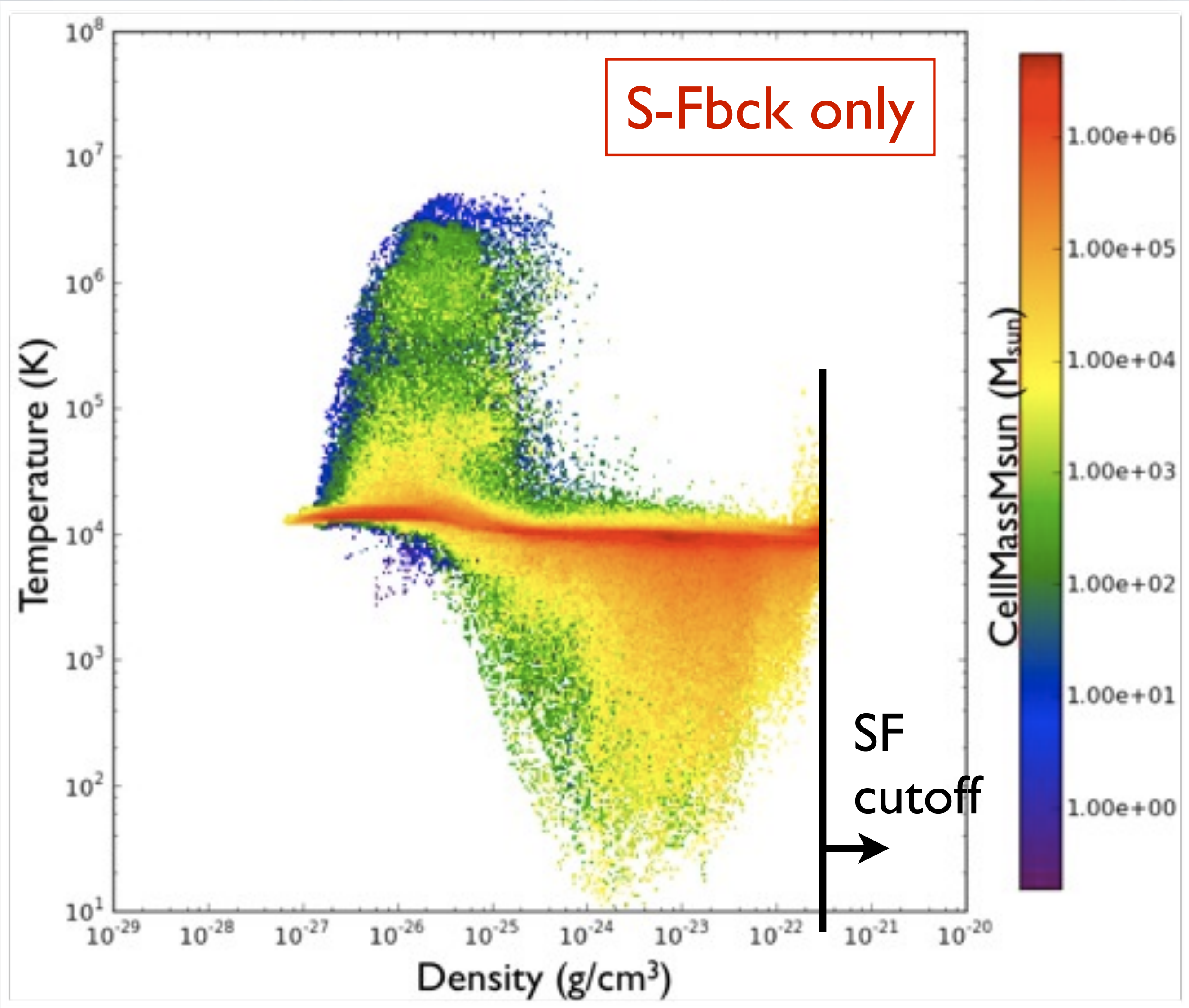


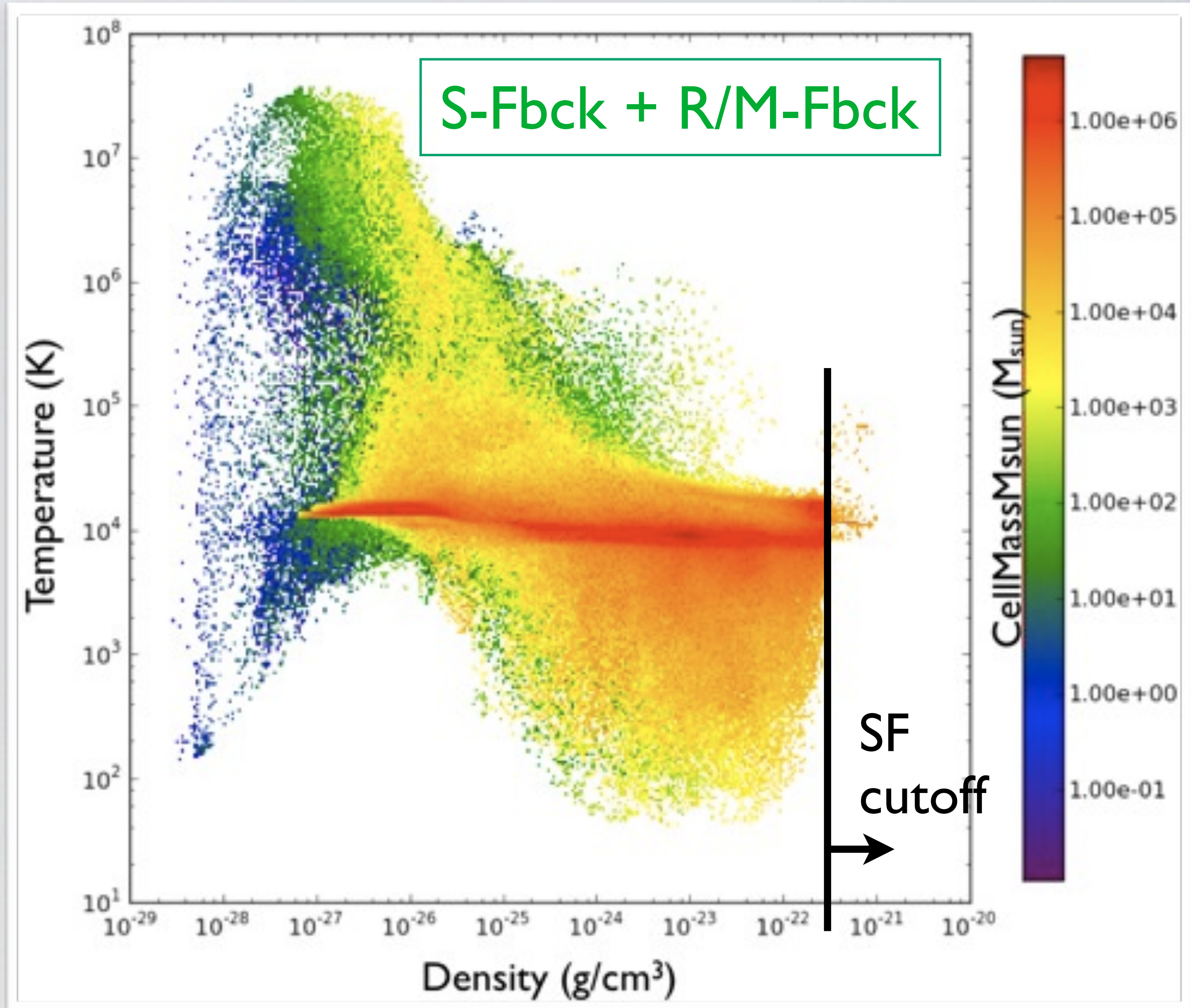
4 kpc centered on a MBH



Density-weighted density proj., 40 kpc, 250 Myrs

Density-Temperature PDF

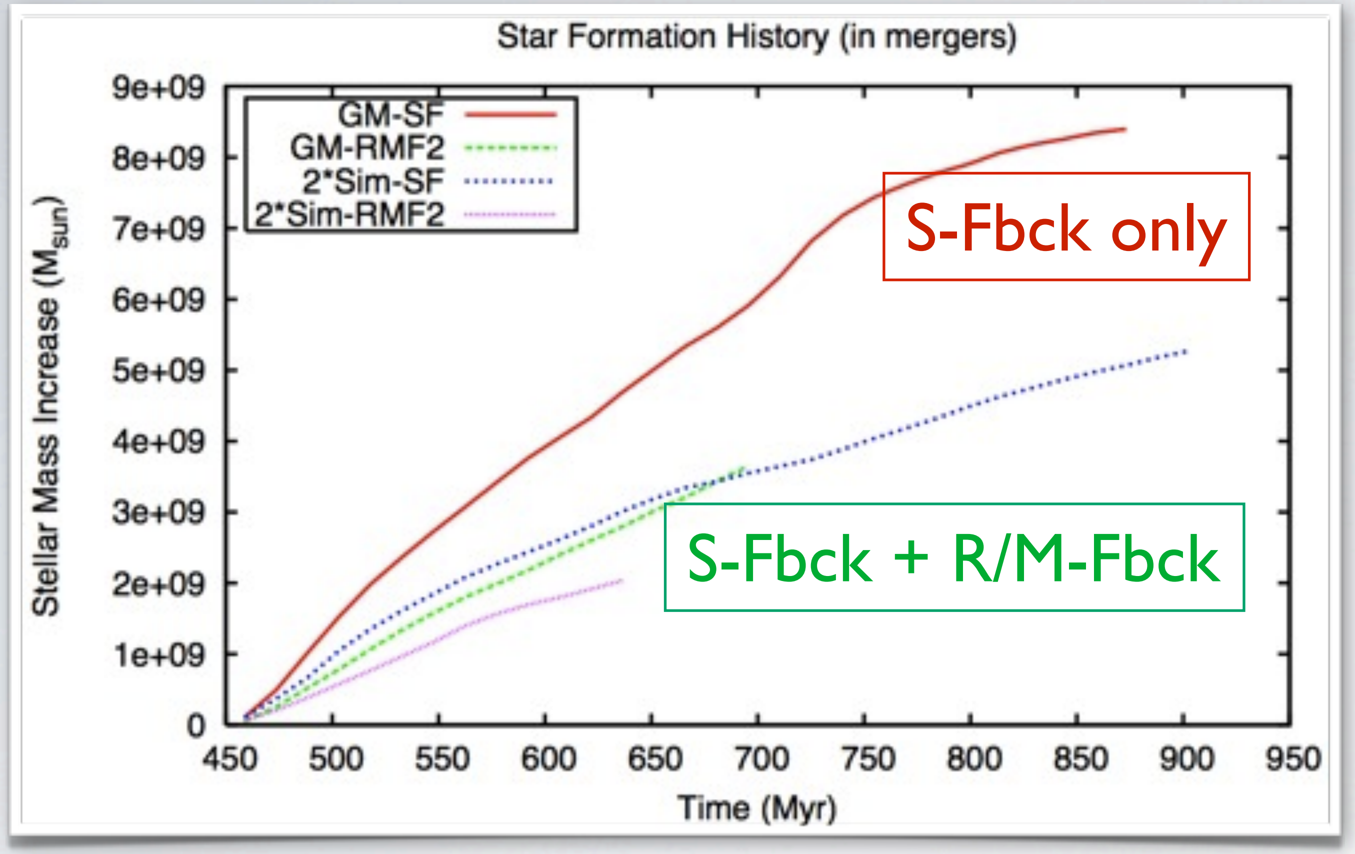




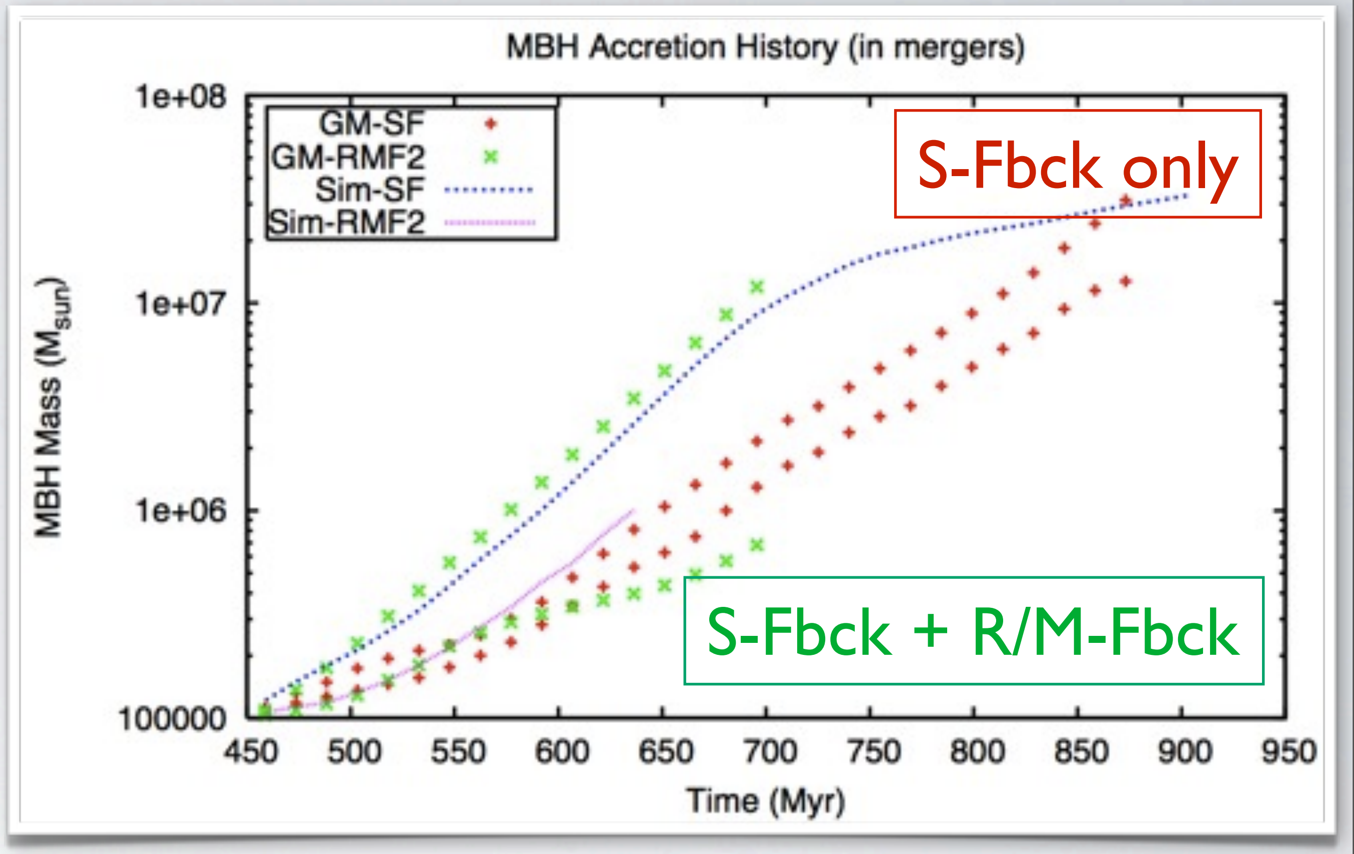
PDF in a 10 kpc sphere centered on one of MBHs

- X-ray radiation significantly changes the ISM, and thus SF
- Hot temperature near a MBH prohibits nuclear star formation

SF and BH Accretion History



SFH (total stellar mass increase)

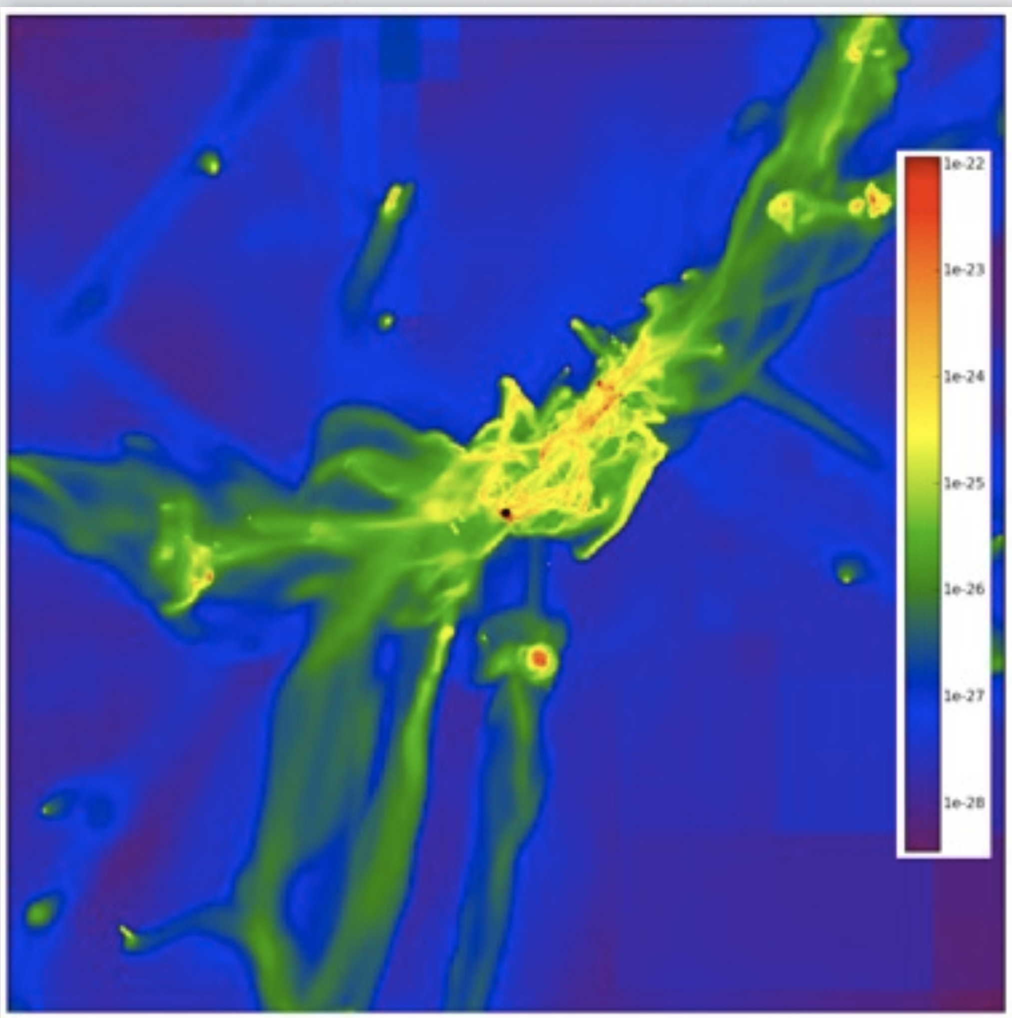


BHAH

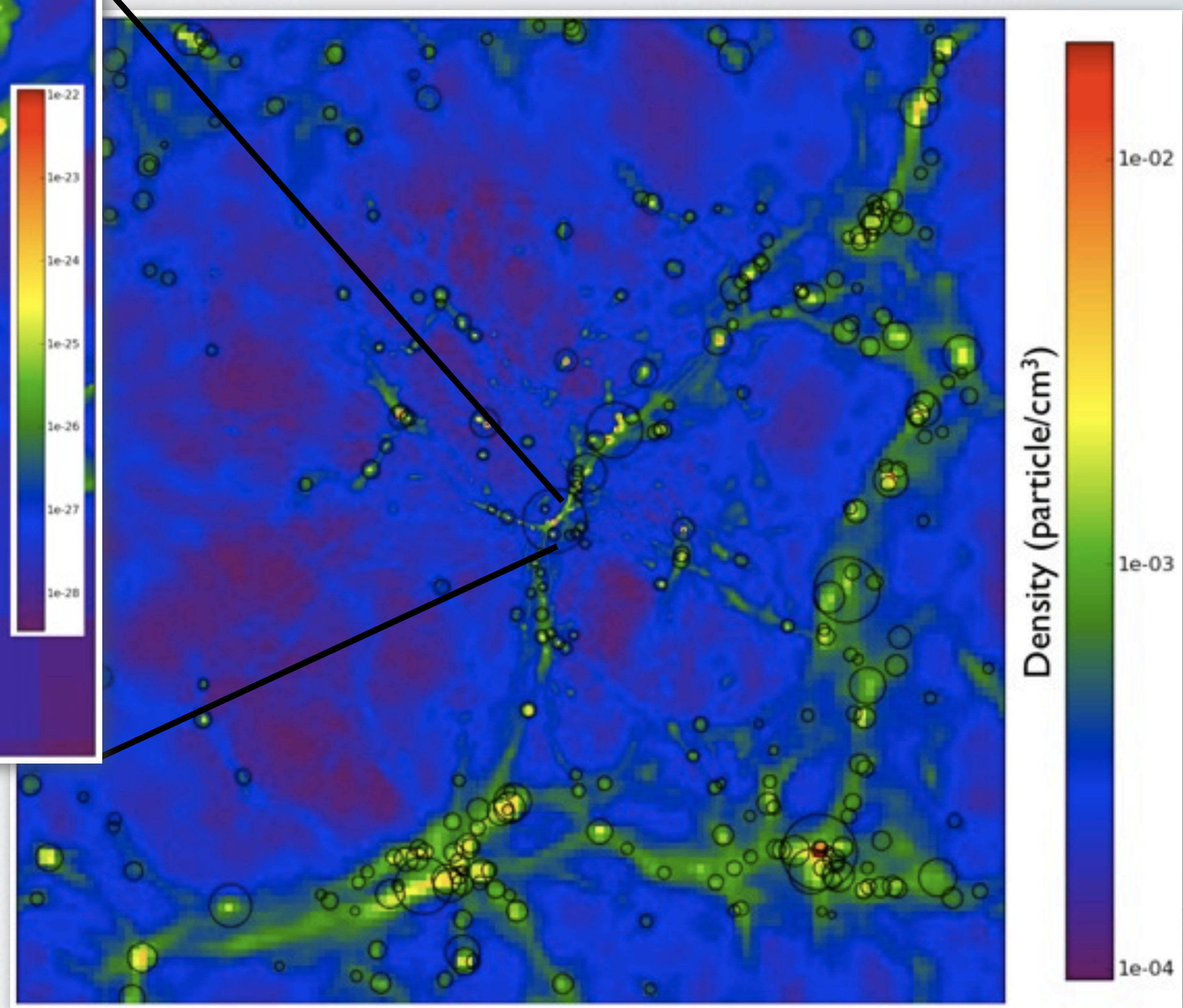
- Star formation rate suppressed by soft X-ray radiation from MBH; more to see as two galaxies start to merge
- Jets do not impact much in regulating accretion as they are mostly perpendicular to gas disks

Cosmological Galaxy Formation at z=3

- A ~10^{12} M_{sun} galaxy selected at z=3 in a low-resolution run
  → insert a 10^5 M_{sun} MBH and restart with 15.2 pc resolution

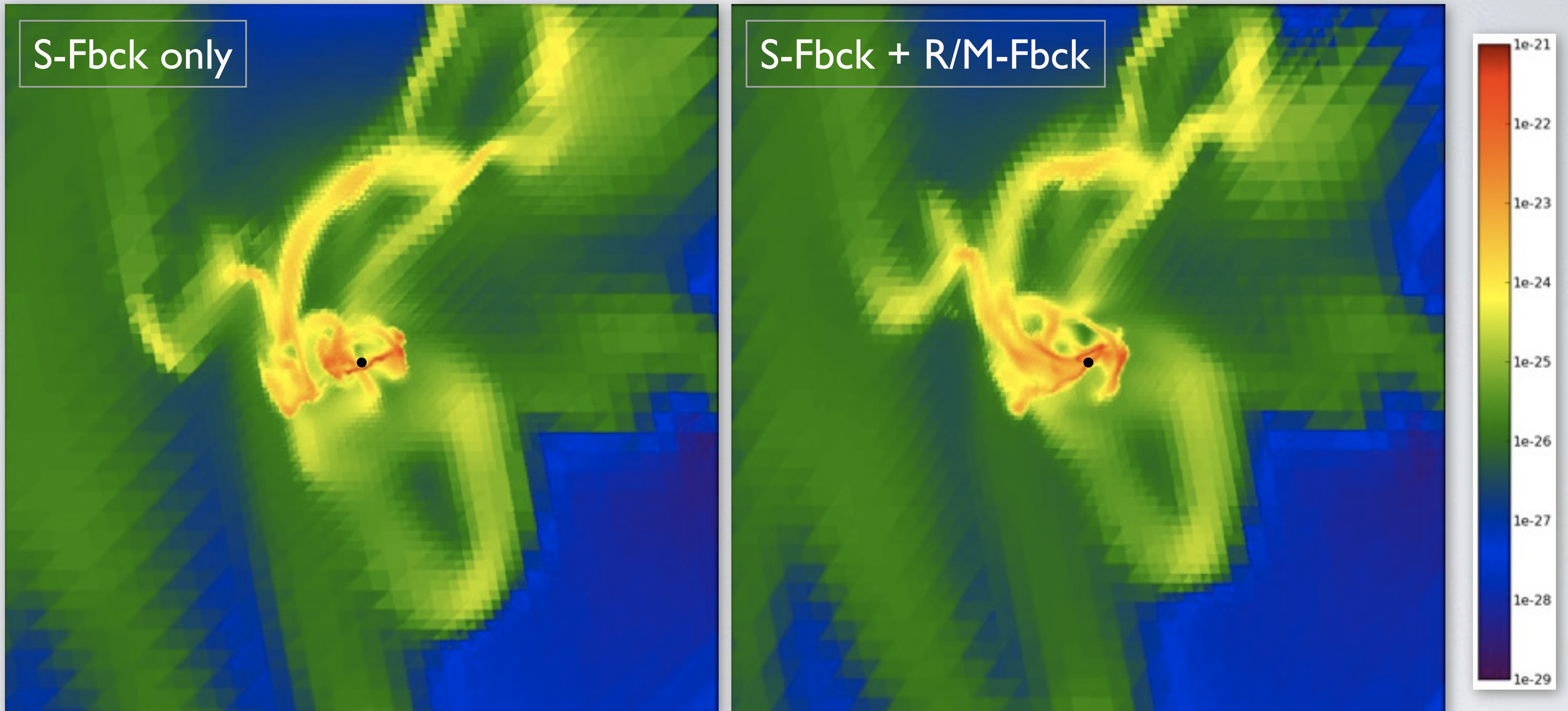


200 kpc centered on a MBH



z=3, Density projection,16 comoving Mpc

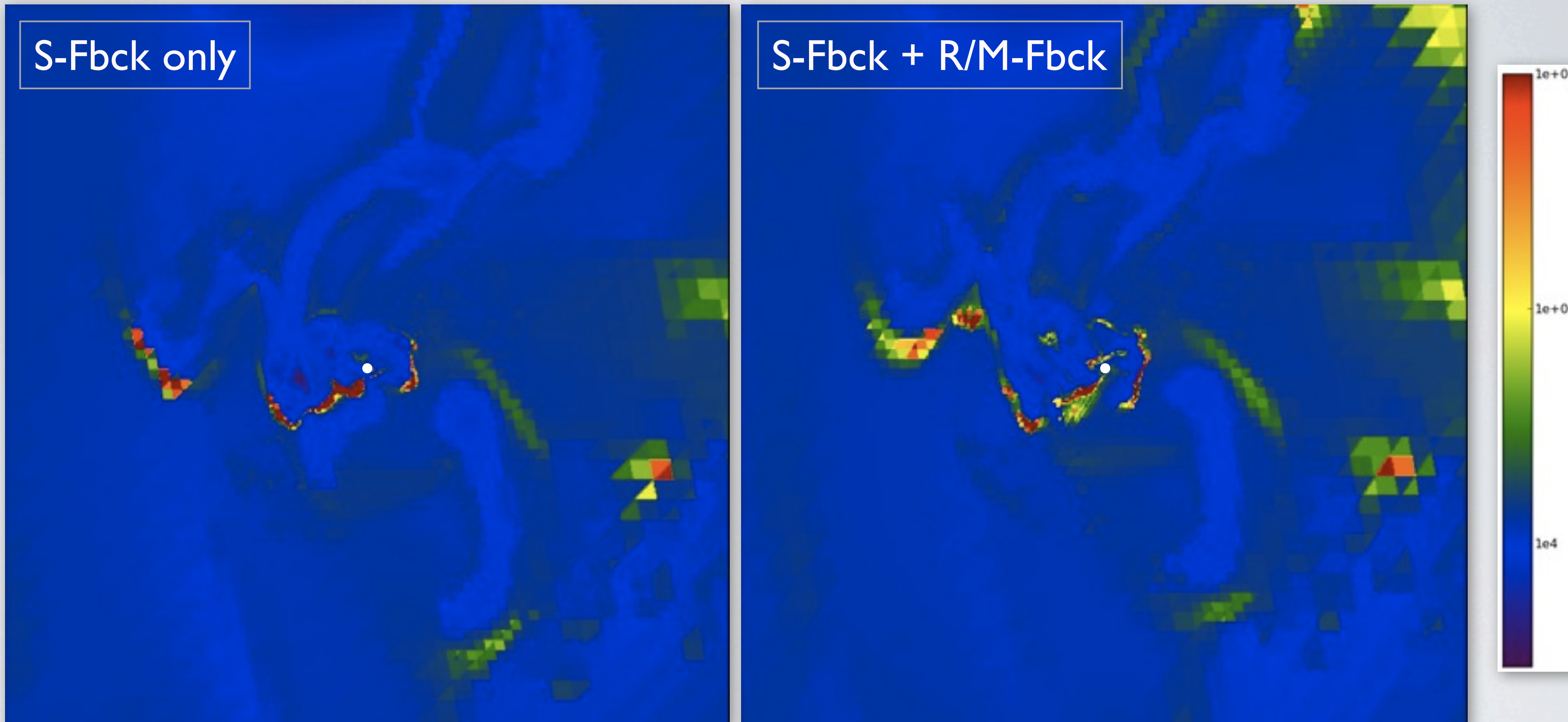
Density Slice



Slice perpendicular to L, 100 Myrs, 20 kpc

- X-ray radiation heats up gas clumps and suppresses SF (probably more efficiently because there is no well-defined disk)

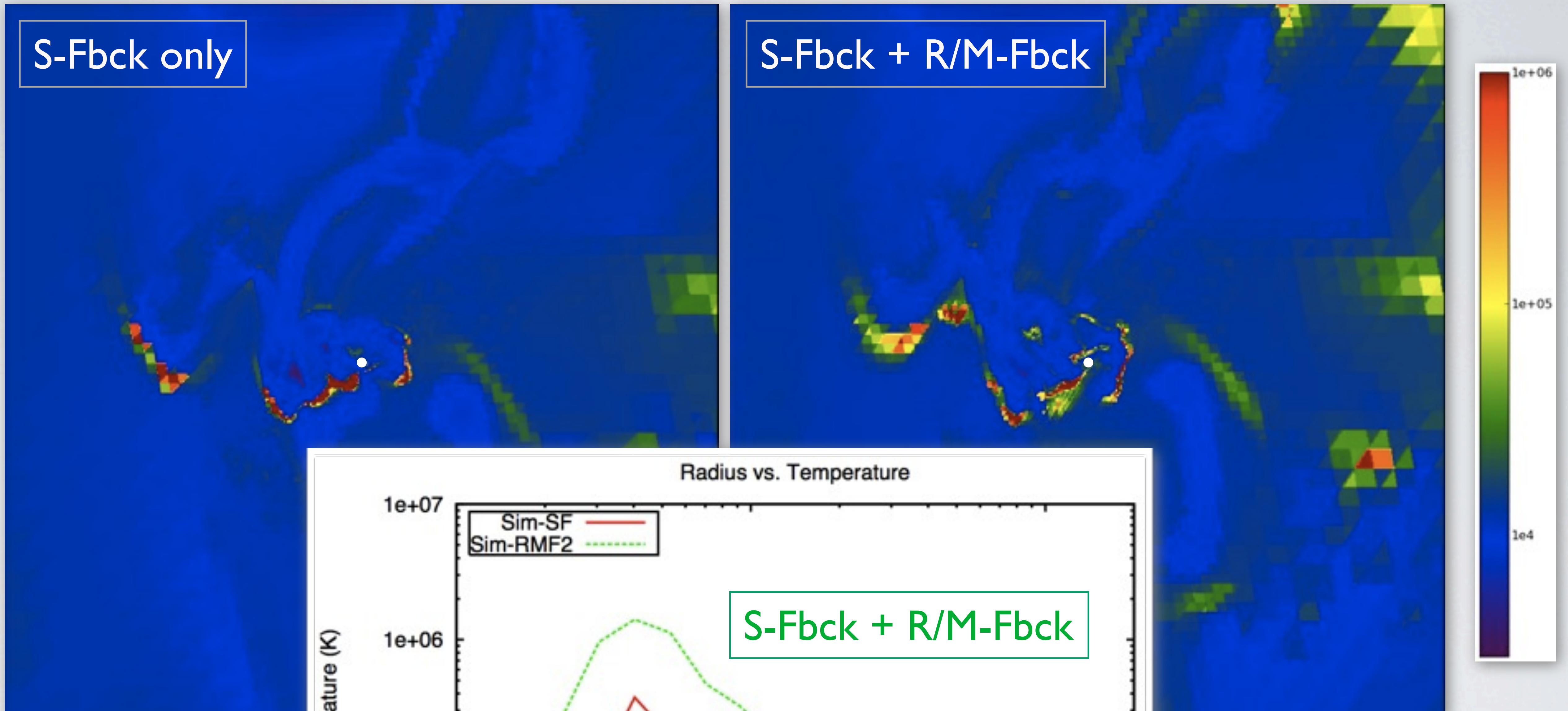
Temperature Slice

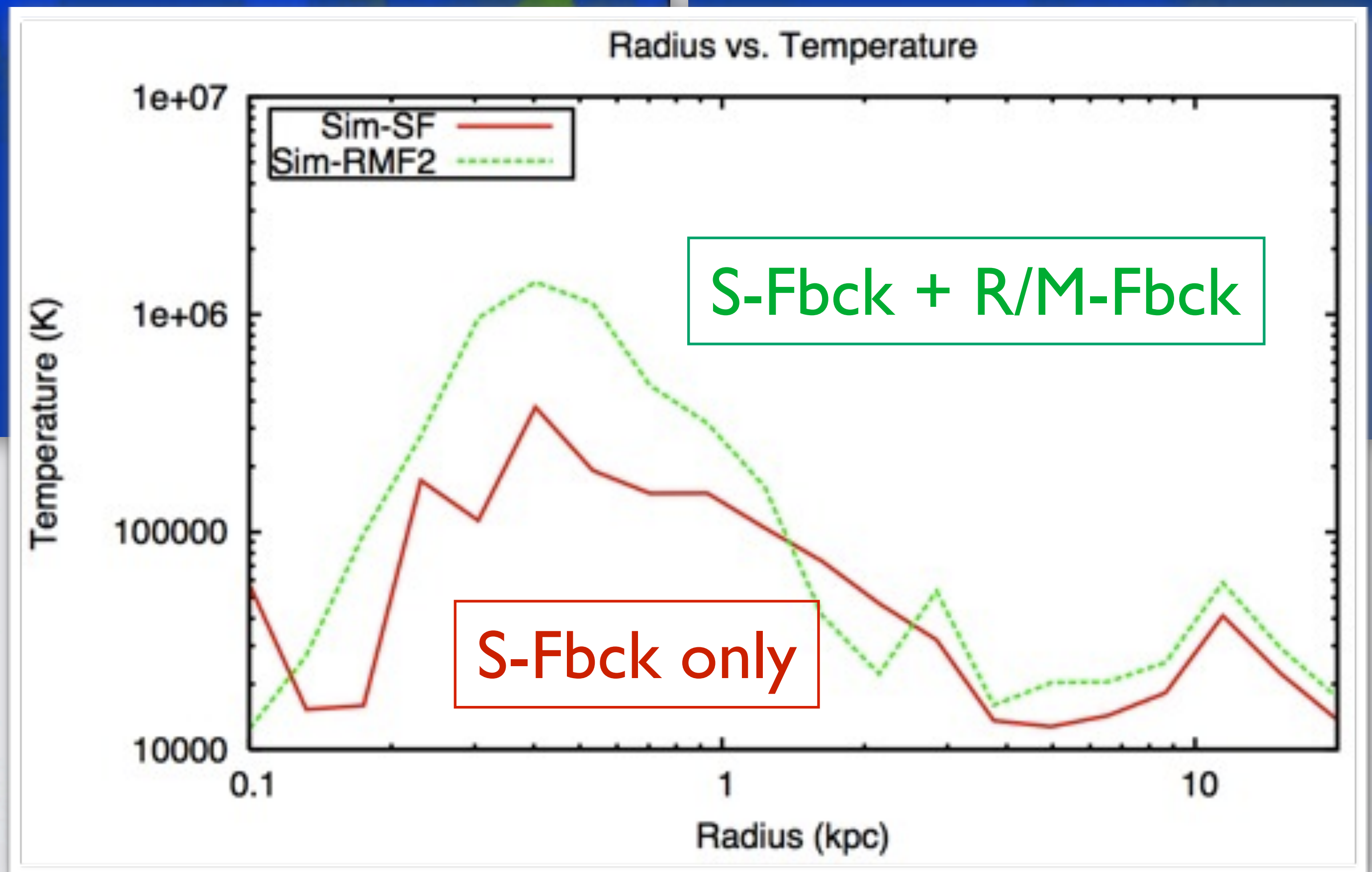


Slice perpendicular to L, 100 Myrs, 20 kpc

- X-ray radiation heats up gas clumps and suppresses SF (probably more efficiently because there is no well-defined disk)

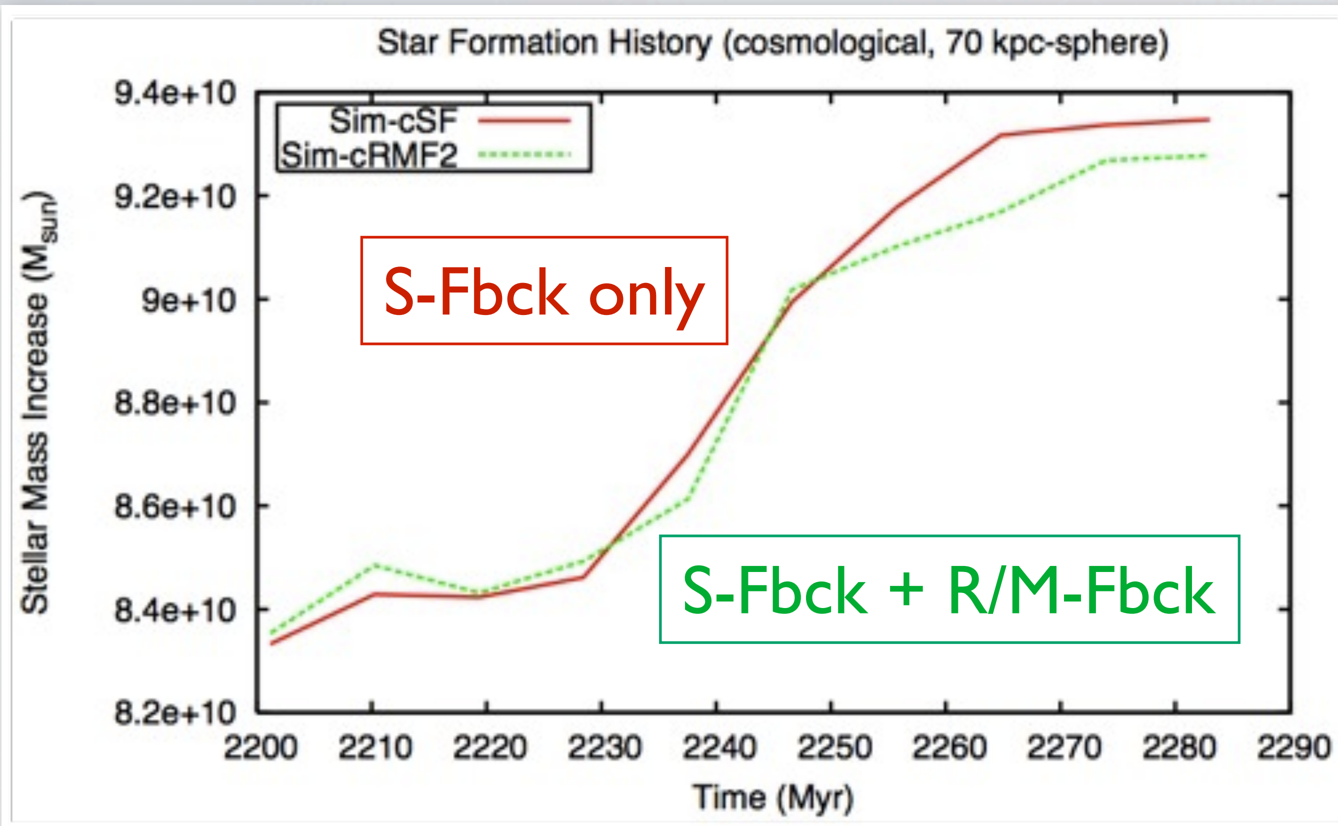
Temperature Slice



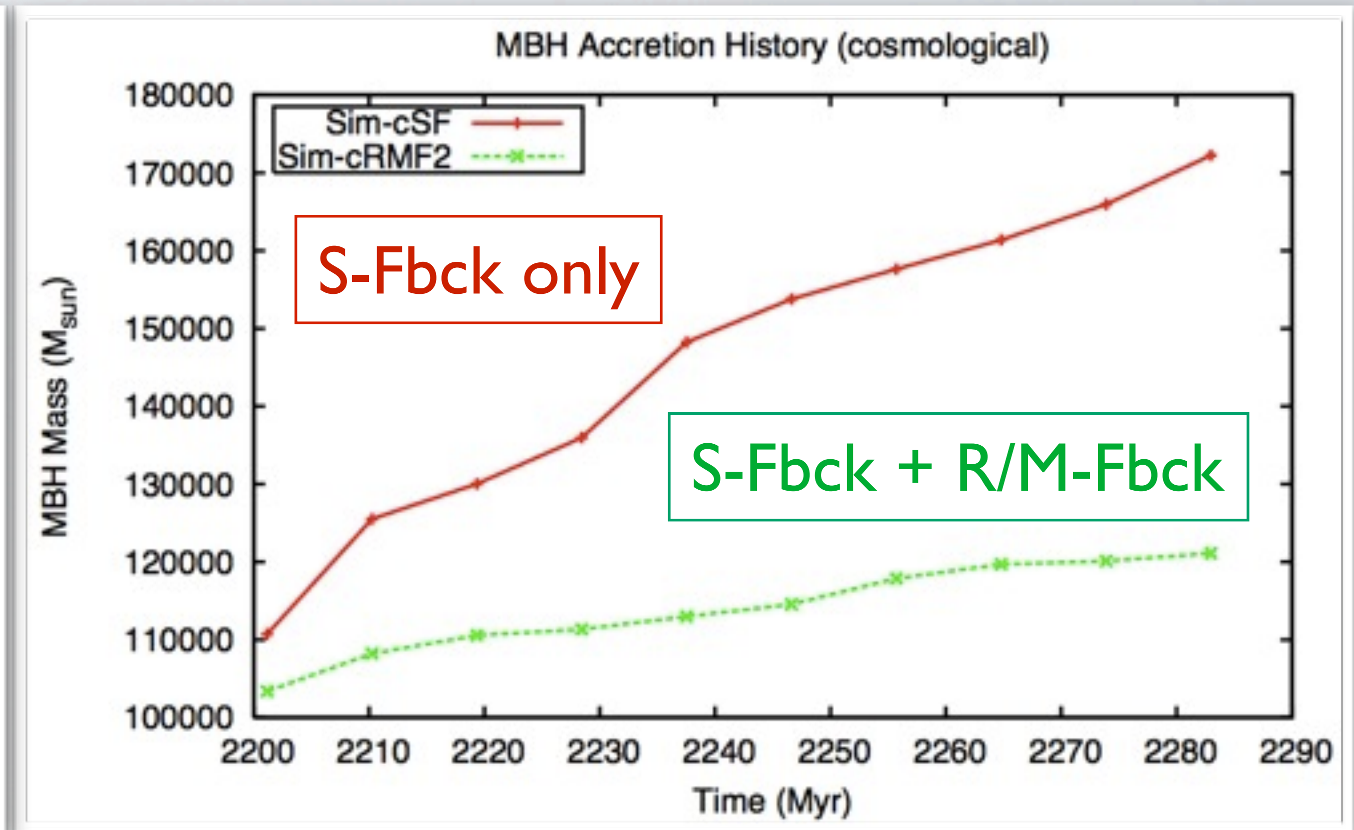


Temperature profile, 20 kpc

SF and BH Accretion History



SFH (in a 70 kpc sphere centered on the MBH)



BHAH

- Radiation also regulates the accretion on to the MBH
- Jets should make more impact with no well-defined gas disk

Conclusion: We are pushing the limit!

- Various components for understanding the physics of galaxy formation are being pieced together in AMR:

  - Proper treatment of MC formation & feedback
  - Proper treatment of MBH accretion & feedback

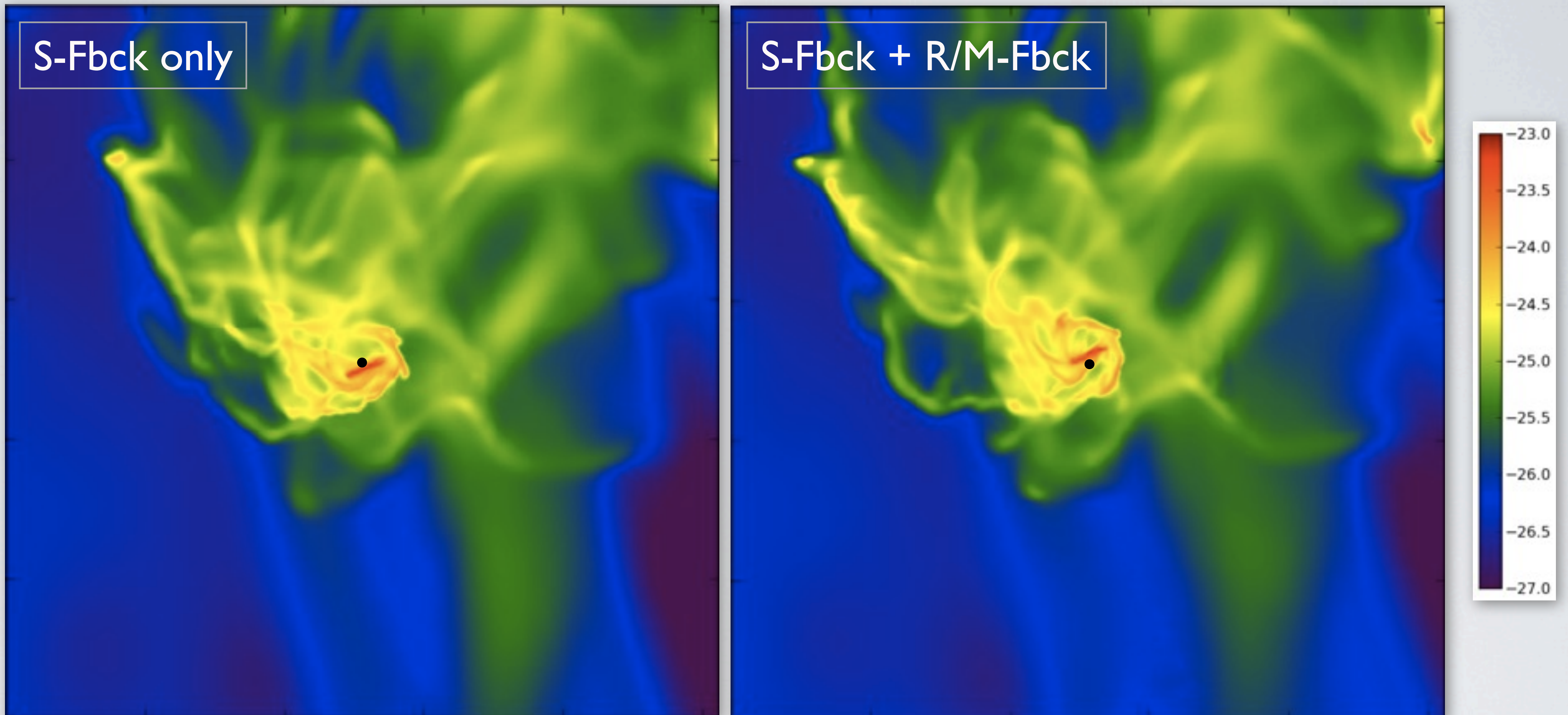
- Preliminary results very encouraging:

  - Stellar and MBH processes in one self-consistent framework
  - Radiation from MBH regulates SF and its own growth
  - Much more to come!

- Kim, Wise, Alvarez, & Abel (2010) in prep.
- Kim, Wise, & Abel (2009) ApJL 694 L123

[Supplemental Slides]

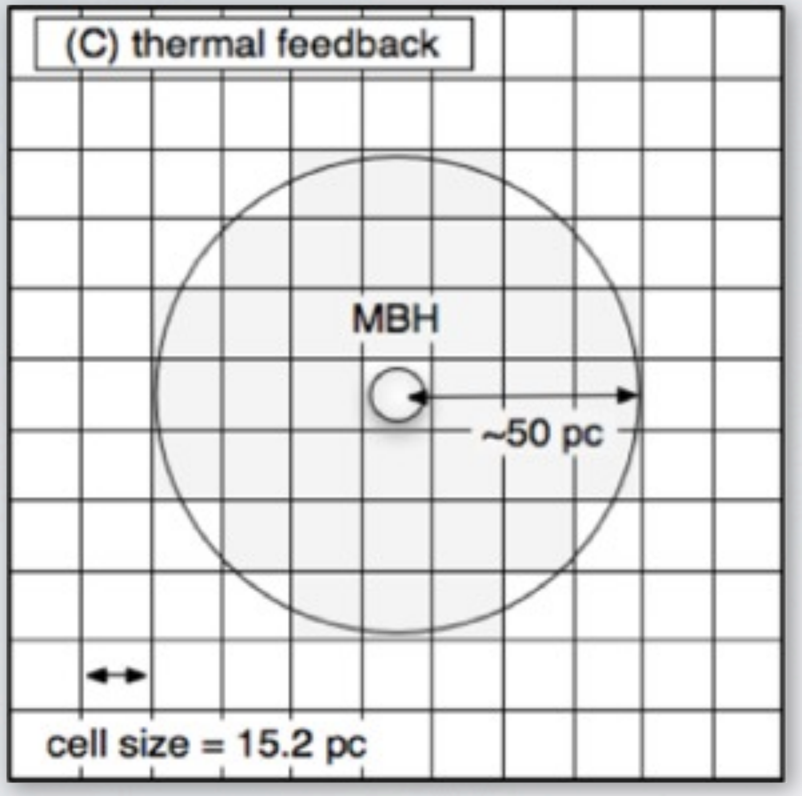
Density Projection Along L

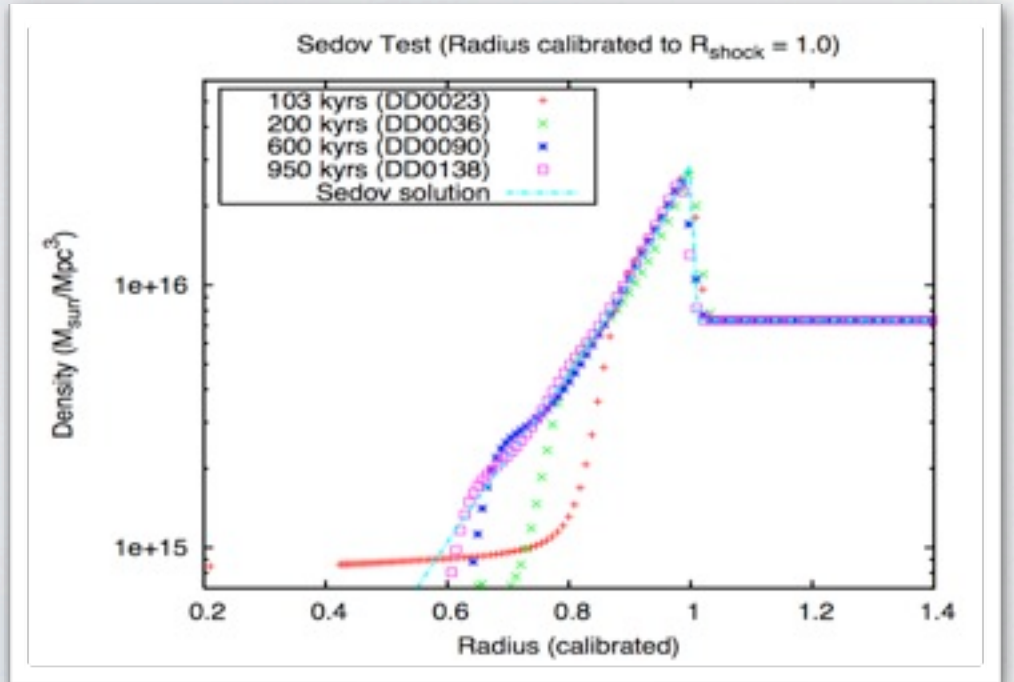


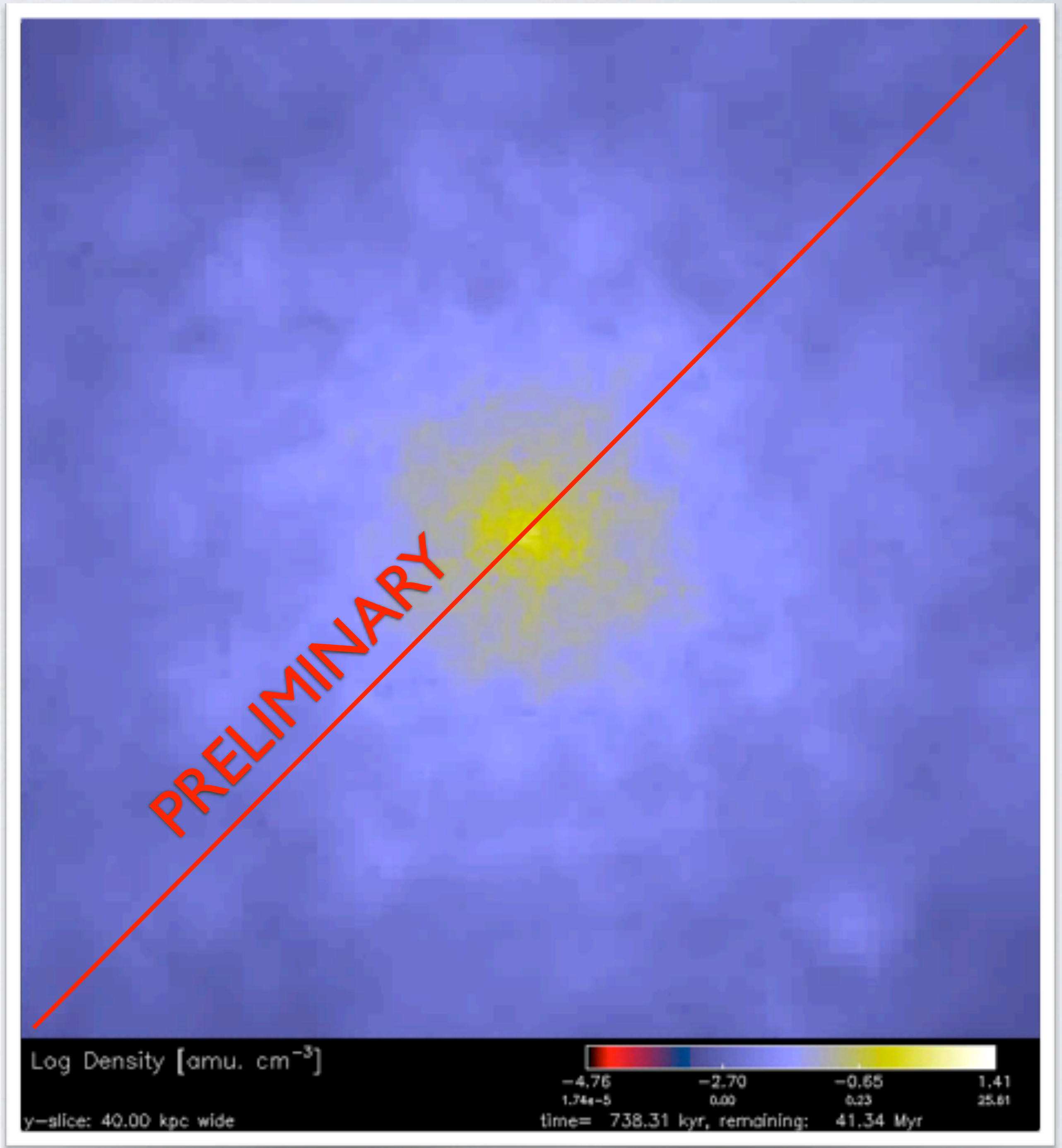
Projection along L, 100 Myrs, 20 kpc

- Too early to compare morphological differences, yet

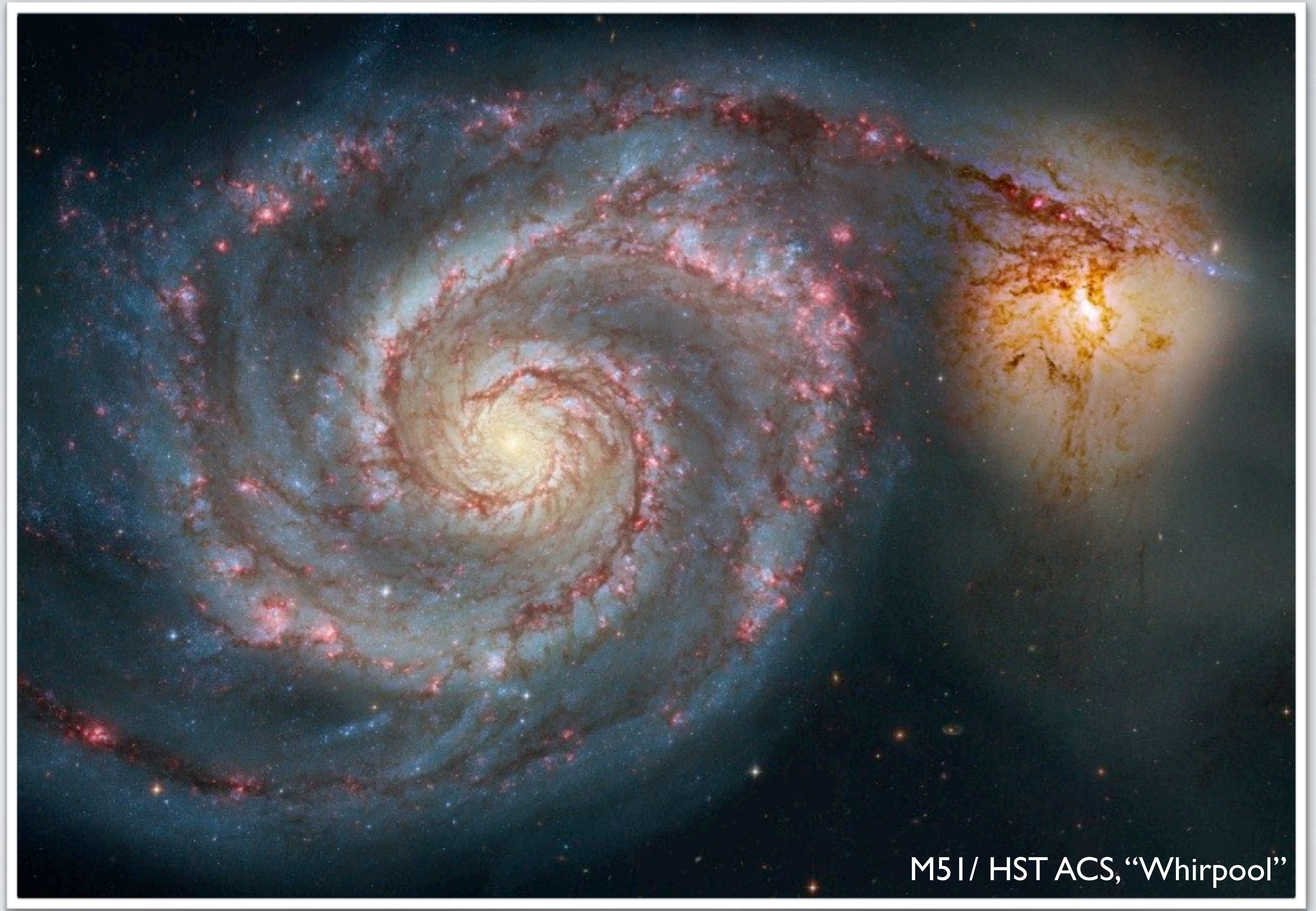
MBH Thermal Feedback



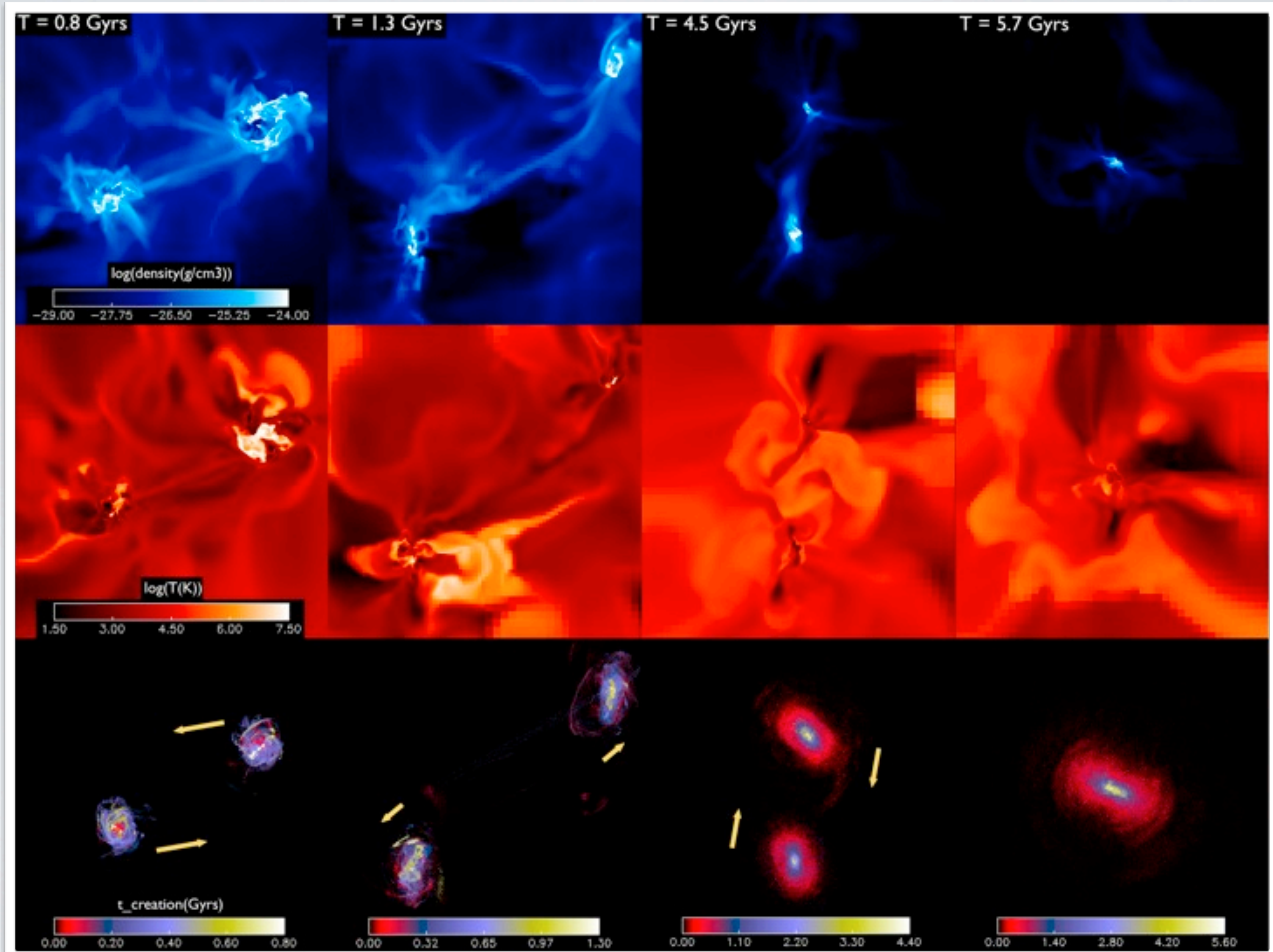




Galaxy = Gas + Stars + MBH + DM, etc.



Merger Sequence



- Kim, Wise, & Abel (2009) ApJL 694 L123